

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
12 July 2001 (12.07.2001)

PCT

(10) International Publication Number
WO 01/50278 A1

(51) International Patent Classification⁷: G06F 13/00 (74) Agents: MALLIE, Michael, J. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(21) International Application Number: PCT/US00/35155 (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(22) International Filing Date: 21 December 2000 (21.12.2000) (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(25) Filing Language: English (26) Publication Language: English

(30) Priority Data: 60/174,362 4 January 2000 (04.01.2000) US 09/538,874 30 March 2000 (30.03.2000) US

(71) Applicant: APPSPPOINT [US/US]; Suite 5, 536 Weddel Drive, Sunnyvale, CA 94089 (US).

(72) Inventors: GEORGE, Jude; 18432 Las Cumbres Road, Los Gatos, CA 95033 (US). BAILEY, F. Ronald; 272 18th Avenue, Santa Cruz, CA 95062 (US). LISOTTA, Anthony; 823 Clarkson Drive, San Jose, CA 95136 (US). MILLER, David, E.; 1325 North State Parkway #17A, Chicago, IL 60610 (US).

Published:

With international search report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/50278 A1

(54) Title: METHOD AND APPARATUS FOR INVOKING A VARIABLE BANDWIDTH EXPERIENCE FOR AN END-USER

(57) Abstract: Techniques allow end users (101A) to view content and use applications requiring varying bandwidth and quality of service/class of service characteristics without the end users' needing to know anything about network technology and provide billing for end-user or content-provider (103) (including advertiser) based on actual network usage measurements or counts. For content delivery/distribution, an apparatus includes an end-user client interface (101) non-technical users to manipulate successfully a network services broker (102A) that manages network resource (both hardware and software, including caching systems), and an application service broker (103A) that manages (directly or through other mechanisms) the distribution source of applications and content. End users can obtain content and use applications needing concurrent human intervention to make these experiences available.

METHOD AND APPARATUS FOR INVOKING A VARIABLE BANDWIDTH EXPERIENCE FOR AN END-USER

FIELD OF THE INVENTION

The field of the invention is related to end-user managed applications/content distribution using electronic networks.

BACKGROUND OF THE INVENTION

Network Service Providers (NSPs) create the physical infrastructure over which all data and voice traffic flows today. They own or lease the equipment that determines the traffic patterns of our global communications infrastructure. This equipment consists of Internet Protocol (IP) routers and switching equipment such as Digital Subscriber Line Access Multiplexers (DSLAMs), Asynchronous Transfer Mode (ATM) switches, and voice switches. Each NSP has authority to control and configure its own equipment. NSPs form peering arrangements with each other to exchange data traffic at Network Access Points (NAPs). The Internet is a conglomeration of NSPs that exchange IP traffic at NAPs. NSPs that only provide Internet services are known as Internet Service Providers, or ISPs.

The customers of NSPs are known as end users. An end user may be an individual at home, a small business person who receives network service at one location, or an employee of a large business or organization that receives network service at multiple locations. In the latter case, the end-user may use the public Internet for traffic between its branch locations, or it may have an intranet -- a dedicated wide-area network provided by its NSP. In all of these scenarios, the end user typically pays a fixed monthly charge for network data services and receives network service on a static (bandwidth fixed) basis.

Application Service Providers or content providers (hereinafter ASPs) typically do not own or manage any portion of the network infrastructure. They are providers of applications and distributors of content. The ASP may have originated the applications and content; often others originate it. The ASPs own or lease servers which are located at or are connected to an NSP, and which are accessible by the NSP's customers. These servers deliver applications and media content to the end users over the NSP's network. Thus, the NSP's network customers can be the ASP's application customers.

In many cases, the relationship between the end user, NSP and ASP is not as cut-and-dried as depicted in the preceding paragraphs. ASPs may, for example, deliver applications and media content to users in the Internet at large. In this common scenario, there is generally no business relationship between the end user's NSP and the ASP, simply because the end user's NSP does not track who the user is contacting via the network service. Thus, there is no method for the ASP to offer a value-added service to the end user, if that value-added service requires an increased level of performance. Likewise, there is no incentive for the NSP to provide an increased level of network service, since there is no compelling application that will both take advantage of it and pay for it.

In all cases, simple or hybrid, although it is ultimately the ASP's task to ensure that its content or networked applications are delivered to the end users with appropriate bandwidth, current conditions do not facilitate this happening with any assurance. Network bandwidth is at a premium and it is not financially or technically feasible for NSPs to provide connections of the requisite level of service to all users all the time. Even as bandwidth becomes more plentiful at lower price points with the deployment of advanced transmission technologies, there is no satisfactory method of making it efficiently available to end users or charging for it appropriately.

There are hundreds of millions of user-initiated application sessions and requests for content downloads from the Internet and from private networks leased or otherwise serving individuals and businesses. However, almost all those sessions and downloads are initiated without end user or content-provider control over the quality of the experience other than a general match-up of typical user need for bandwidth and bandwidth provided by an NSP and/or ASP.

In digital applications and content delivery, much of the quality of the user experience is determined by the availability of bandwidth that is optimal for the specific experience. In the case of relatively low bandwidth applications, such as simple word processing, not much bandwidth is required, and even a low-speed, dial up connection such as a 14.4kbs modem will suffice. In the case of high-bandwidth applications and content, such as graphics-rich materials, much more bandwidth is required. High quality video experiences, for example, require 6-8 megabit transmissions.

Two other determinants of the quality of end users' experiences are largely influenced by bandwidth considerations. Latency (delay) in transmission can increase if too many users

are "on" the network at the same time. Jitter (statistical variation in latency) can also increase if there is too much congestion in network traffic.

Traditionally, end users or their employers enter service agreements with network service providers for enough bandwidth to meet most of their needs most of the time. As a result, they pay on a steady state basis for a specific amount of bandwidth that is sometimes more than what they need (for simple applications like word processing), and sometimes less than what they need (for high-resolution video distribution). Traditionally, the idea is to strike a balance between end user needs for bandwidth and the cost of service, which in turn is charged on a steady-state basis (e.g., \$19.95/month for unlimited Internet access at 56kbs service, \$70.00/month for 768kbs service, etc.).

Although this approach strikes a reasonable balance of "average need/reasonable price" for the "average" use of networked resources, it greatly inhibits the range of services experienced by and provided to end users.

The current rather static method of invoking and charging for bandwidth involved in applications usage and content download is likely to impose increasingly unsatisfactory tradeoffs as new services and technological capabilities come on-line.

Another problem with the current state of the art is the fact that content providers do not effectively regulate admission to experiences to the number of users who can effectively access them under conditions permitting the experiences to be properly enjoyed. Instead, if too many users establish connections to the experience, it degrades to the point where no one experiences it as originally intended. This is similar to a movie theater continuing to sell tickets to a show and admit people even after all the seats in the theater have all been filled.

Another problem with the state of the art is that network providers "over subscribe" lines serving DSL customers. That is, they sell more bandwidth than is available on the expectation that customers will not normally all be logged on and interacting with the Internet at the same time. When there is a popular event, performance degrades for all users. Over subscription occurs for all types of customers but is particularly severe in the case of consumer connections.

Another problem with the current state of the art is that Internet and other network advances affecting end users' ability to enjoy experiences with varying performance characteristics-- including without limitation advances in end-user bandwidth options, network upgrades, improvements in network management systems, including middleware improvements, IP class-of-service standardization, development of applications, and both

technical and legal issues involved in multi-jurisdictional collaboration (voluntary or enforced through government action)—all proceed at an uneven pace. Although interrelationships among such factors are not uncommon, there is no guarantee that an advance in one area will generate a corresponding improvement in another.

Current methods of coping with these problems are unlikely to provide the kind of bandwidth control and granular pricing required.

Current conditions do not ensure that its content or networked applications are delivered to the end users with appropriate bandwidth. Network bandwidth is at a premium and it is not financially or technically feasible for NSPs to provide connections of the requisite level of service to all users all the time. Even as bandwidth becomes more plentiful at lower price points with the deployment of advanced transmission technologies, there is no satisfactory method of making it efficiently available to end users or charging for it appropriately.

Traditional approaches to circumvent these problems have produced limited success. They include the following methods and suffer the indicated shortcomings:

- 1) The ASPs and NSPs must set up network service levels in advance so that users (corporate and consumer) pay regularly for bandwidth used only some of the time. The bandwidth, although greater than what is often needed, may not be adequate for some applications/content that requires still higher performance characteristics. This typical scenario affords a crude match up between end user bandwidth and budget considerations, as outlined above, and retards development of creative user experiences. The current art thus inhibits revenue growth for service, content, network, and hosting providers.
- 2) Situational modification of service levels by providers, conducted by humans through direct interaction with the network components and application/content sources. This method is very labor-intensive and expensive; it suffers from the competition for technicians' attention, chances for error, and opportunities for delay. The result of this approach is typically to provide the end user with higher bandwidth for a longer period than is actually required, or for too short a period if the experience needs to be extended. Because it is so labor intensive, it confronts the network provider with an unhappy choice between charging much higher prices (thereby inhibiting demand) or suffering low margins (thereby hurting the providers' profitability)
- 3) Use of human interaction with middleware (usually hardware/software combinations or software which works with other vendors' equipment) within the network.

Middleware is used to impose "policies" or designated special treatment for certain types of traffic, and to calculate discrete usage information. Such hardware and software middleware products, properly used, can improve network utilization and efficiency, but they do not address the need for an automated end-user invocation of bandwidth-variable experiences, or any means for end users to ascertain for themselves whether the requested service levels are being provided. This method also fails to effect the necessary collaboration between NSP and ASP roles required to ensure a successful end user experience.

4) Regionally hosting applications/content at the NSP or NSP-proximate ASP, and thus closer to the end-user. The advantage of this technique is to by-pass the uncertain performance characteristic of the Internet. However, the last leg of the delivery network remains the static bandwidth-fixed connection described above with its inherent limitations and inefficiencies.

5) Caching content at the NSP or proximate ASP to simulate direct delivery from the originating source without Internet- or other network- imposed degradations in quality. This technique enjoys the same advantages as regional/local hosting, but can be expensive for large applications or units of content. Also, this method does not provide situational control to the end users and to those who would provide paid-for experiences to end users.

6) Another method being advanced is to provide virtually infinite bandwidth to all users by developing economies of scale to the point where affording end users massive amounts of bandwidth will come with a price tag so low that cost will no longer be a consideration for end users and their employers. This is an attractive vision, but there is no evidence that this approach will succeed. There is also the danger that, even if bandwidth costs approach zero, richer and richer applications will be developed (holographics, three-dimensional faxes), which will drive bandwidth requirements still higher. And even under a free-bandwidth scenario, there will remain the need to charge for application/content experiences which vary in complexity and value and which must be transmitted between the end user invoking the experience and the ASP provider.

Thus each of the above-listed approaches embodied in the current art fails to address certain needs.

SUMMARY OF THE INVENTION

A method comprising receiving a request for a bandwidth-varied experience (BVE) from a client, and one or more service providers dynamically providing services to satisfy the BVE request.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

Figure 1A illustrates a block diagram of an architecture depicting client 101A associated with client host 101 coupled to ASP 103.

Figure 1B is a more detailed block diagram of one embodiment of an architecture for allowing an end user to invoke varying bandwidth, and/or quality of service or class of service on an application basis.

Figure 2A is a flow diagram with one embodiment of a process for an end user to obtain and use a single purpose client.

Figure 2B is a flow diagram of one embodiment of a process illustrating the end user receiving an enhanced experience or rain check applet.

Figure 3 is a flow diagram of one embodiment of a process for obtaining high performance with an end user client built into an application with a performance level option.

Figure 4 is a flow diagram of one embodiment of a process for obtaining higher performance with an end user client built into an application with a higher performance level also built in to the application.

Figure 5 is a flow diagram of one embodiment of a process for an NSP broker to determine the readiness for the bandwidth varied experience (BVE) from, for example, NSP elements, middleware and/or caches.

Figure 6 is a flow diagram of a process illustrating operation of the end user client proxy performing traffic monitoring.

Figure 7 is a flow diagram of a process illustrating operation of end user client proxy to determine the NSPs and application tone on behalf of the client.

Figure 8 is a flow diagram of a process illustrating an application using a library on the client host to invoke an enhanced service.

Figure 9 illustrates a desktop view of one embodiment of the applications player's window.

Figure 10 illustrates the launch pane of one embodiment of the applications player.

Figure 11 illustrates one embodiment of a collapsed view of the applications player.

Figure 12 illustrates another pane of the applications player.

Figure 13 illustrates one embodiment of the launch of an experience.

Figure 14 illustrates one embodiment of a summary pane that sets forth a log of transactions related to the BVE.

Figure 15 illustrates an exemplary web page from which the applications player may be launched.

Figure 16 is a flow diagram of a process illustrating a user initiating an experience at a service provider's web site.

Figure 17 is a flow diagram of one embodiment of a process by which an end user learns of available experiences by registering at ASP's web site.

Figure 18 is a flow diagram of one embodiment of a process by which a user learns of available NSPs by registering at E-Hub.

Figure 19 is a flow diagram of one embodiment of a process by which an end user obtains a sample of enhanced service.

Figure 20 is a flow diagram of one embodiment of a process by which a user selects content quality level.

Figure 21 is a flow diagram of one embodiment of a process by which a proxy automatically initiates an option for higher service.

Figure 22 is a flow diagram of one embodiment of a process by which an end user invokes an enhanced service paid for by an advertiser.

Figure 23 is a flow diagram of a process illustrating roaming by the end user.

Figure 24 is a block diagram illustrating the broker requesting the OSS software to activate resources in the network such as, for example, DSLAMS and ATM switches.

Figure 25 is a flow diagram illustrating one embodiment of a process by which an NSP broker effects service through DSLAM.

Figure 26 is a flow diagram of one embodiment by which an NSP broker effects service in a wireless network.

Figure 27 is a flow diagram of one embodiment of a process by which a broker negotiates bandwidth in IP over sonet network.

Figure 28 is a flow diagram of one embodiment of a process by which an NSP broker negotiates bandwidth in analogue or digital cable network.

Figure 29 is a block diagram illustrating the broker managing policy in an IP network through a policy server.

Figure 30 is a flow diagram of one embodiment of a process by which a broker negotiates enhanced service transmission through Ethernet, gigabit Ethernet, or fast Ethernet facilities.

Figure 31A is a block diagram of one embodiment of a network environment.

Figures 31B-F are block diagrams illustrating various network arrangements.

Figure 32 is a block diagram of an exemplary computer system.

DETAILED DESCRIPTION

A method and apparatus for invoking a variable bandwidth experience is described. In the following description, numerous details are set forth. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks,

CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory ("ROM"); random access memory ("RAM"); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

General Overview

A method and apparatus for allowing an end user to obtain an additional network service (e.g., a value added network service) dynamically without contacting an NSP or ASP directly is described. The added service may be a higher bandwidth network connection from one location to another location, or it may be an improved connection along with an application (from an ASP) to take advantage of the improved connection.

NSP and ASP brokers (e.g., NSPs and ASPs with installed brokering software) are responsible for orchestrating the changes in order for an end user to obtain use of the additional service. In one embodiment, one of these brokers also conducts the billing transactions that may take place.

Figure 1A illustrates a block diagram of an architecture depicting client 101A associated with client host 101 coupled to NSP 102. NSP 102 has an NSP broker 102A to enable the end user of client host 101 to have value added services. NSP 102 is coupled to ASP 103, which has ASP broker 103A that enables the end user of client host 101 to have value added services.

In one embodiment, an applications player, described in greater detail below, is an application which resides on the end user's computer system (e.g., desktop handheld, etc.) and displays a window representing the network services and/or applications that are available to that end user. The end user may use the applications player to select one or more of the available network services and applications.

If a service provider has a broker and offers the capability for their customers (i.e., the end users) to initiate BVEs, they are said to offer applications tone. Applications tone describes the readiness of the network and networked application(s) to accept end user invocation of services, such as, for example, services requiring different performance levels. In other words, application tone, when applied to service providers, refers to the technical readiness of a service provider to dynamically provide services by reallocation or enablement of network resources (e.g., bandwidth, bandwidth-affected performance characteristics). Application tone, when applied to applications, refers to the fact that the application has been written in such a way as to make a request for services that require dynamically allocated or enabled network resources (e.g., bandwidth, bandwidth-affected performance characteristics).

There are a variety of scenarios in which end users may desire to invoke services. In one embodiment, when desiring to use the applications player, the end user clicks the applications player on his desktop and determines if he has applications tone. The end user selects and requests a service or application. If the service is provided by his local NSP, or by an ASP with whom he already has a service contract, the end user launches the service or application and is billed automatically. If the end user requests a service, content, or application from a remote provider with whom he has not yet registered, he is prompted to either pay for the service (e.g., via entry of credit card or other financial information) or to register to qualify for a sponsored experience. The brokers for all involved parties (NSPs and/or ASPs) negotiate the financial transactions required to ensure that the NSPs providing the added network service are compensated either by the ASPs who are selling the application or the media content, or by the end user enjoying the service. The service is then delivered to the end user.

In one embodiment, at each NSP, the broker uses the NSP's existing Operational Support Software (OSS) to perform the operations to supply added network services for individual user connections. The broker may also integrate with the NSPs existing billing system to allow the NSP to charge back the added services to ASPs, employers, merchants, advertisers, and/or end users, if desired. To allow the network devices to deliver the requisite

service levels, standard mechanisms, such as, for example, DiffServ, RSVP, and MPLS, are employed. In an IP network with an ATM infrastructure, such as most DSL environments, the individual ATM virtual circuits may be managed using specific ATM classes of service. In a pure IP routed network, DiffServ and vendor-specific policy managers may be employed.

In order for an end user's network connection to be enabled end-to-end, each NSP is enabled. Each network element in the path of the network connection also sets aside the resources to ensure that the network connection is of sufficient quality to deliver the user's service. Thus, individual network elements are configured to support these service level or are activated at the service level.

Figure 1B is a more detailed block diagram of one embodiment of an architecture for allowing an end user to invoke varying bandwidth, and/or quality of service or class of service on an application basis. Referring to Figure 1, an end user, using the end user client 100 requests a bandwidth-varied experience (BVE), referred to herein as the BVE request 130. "Bandwidth varied experience" (BVE) refers to experiences for which bandwidth is actually varied, as well as experiences requiring existing bandwidth to be consistently maintained and experiences requiring bandwidth-related quality characteristics, such as, for example, latency and jitter, to be enforced. The interface for end user client 100 to make the request may comprise a JAVA applet, thin client, plug-in, or application. The BVE request may include one or more BVE-associated options presented to the user. These options may include being able to receive a lower bandwidth or defer receipt of a particular service until later. The request is transmitted through end user network interface 120, which forwards the request to the NSP application tone broker 210 or ASP broker. In one embodiment, end user network interface 120 may be a DSL modem, cable modem, wireless modem, channel service unit (CSU), or other interface. End user network interface 120 may forward the request through the elements, in-band (e.g., NSP elements such as DSLAM, ATM switches, routers, etc.), or alternatively out-of-band, as for example, through a wireless palm device signal transmission.

NSP application tone broker 210 requests the requisite bandwidth from NSP elements 200. In one embodiment, NSP elements 200 include NSP caching resources. NSP application tone broker 210 may request the requisite bandwidth either directly or through NSP provisioning/activation middleware 220. NSP elements 200, either directly or through NSP provisioning/activation middleware 220, acknowledge availability of and secure the requisite bandwidth or report failure of the request when adequate bandwidth is unavailable. If the request fails, the NSP broker 210 reports the unavailability of the BVE to end user client

100 through end user network interface 120. The report may include available options and/or alternatives.

If adequate bandwidth is available within the NSP domain, NSP application tone broker 210 requests the BVE from ASP application tone broker 300. ASP application tone broker 300 requests the BVE from ASP elements 310 using BVE request 130. If the ASP content/application source 320 and/or ASP elements 310 report that the experience is available, that fact is reported back to the NSP application tone broker 210, and the experience is transmitted to the user through NSP elements 200, which provide performance reports to NSP application tone broker 210. These reports may be issued directly to NSP application tone broker 210 or through the NSP provisioning/activation middleware 220 and acknowledge that NSP elements 200 have provided the requisite performance or that they have failed to provide the requisite performance.

If the BVE is successfully delivered to the end user, NSP application tone broker 210 requests usage information from NSP billing middleware 230. Middleware 230 may communicate directly with the NSP elements 200 or with the NSP provisioning/activation middleware 220. NSP elements 200 or provisioning/activation middleware 220 provide the usage report to NSP billing middleware 230, which forwards the information to NSP application tone broker 210. NSP application tone broker 210 forwards the usage information to NSP billing system 240, which creates a bill 153 through the NSP's billing method 250. NSP billing system 240 may also provide a billing report 152 to NSP application tone broker 210, which submits that report to end user client 100 through end user network interface 120. Alternatively, the bill may be forwarded to the sponsoring organization (e.g., advertiser, etc.).

Figure 2A is a flow diagram with one embodiment of a process for an end user to obtain and use a single purpose client. The process may be implemented using processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 2A, processing logic begins with an end user starting a browser to initiate an end user request (processing block 201). Then the end user goes to an ASP website (processing block 202). At the ASP website, the end user downloads a player (processing block 203) and the end user launches the player (processing block 204).

Once the player has been launched, processing logic initiates a request for bandwidth necessary for a particular experience (processing block 205). In response to the request, processing logic tests whether the bandwidth is available (processing block 206). If the bandwidth is not available, processing logic provides the end user with a rain check or

provides the end user with optional return times (processing block 207). If the bandwidth is available, processing logic provides the available bandwidth to the end user (processing block 208).

Figure 2B is a flow diagram of one embodiment of a process illustrating the end user receiving an enhanced experience or rain check applet. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 2B, the process begins by multiple end users initiating requests for enhanced experience (processing block 211). The request may correspond to, for example, movies, access to trades for a hot securities issue, or admission to participate in a popular auction event. As each NSP-brokered request is received, the ASP broker determines if resources are available for that instance of user experience (processing block 212). Then the ASP broker determines whether there are resources available (processing block 213). If not, then the end user gets a rain check applet, rebroadcast schedule (processing block 214). If resources are available, then admission and/or the enhanced experience is provided to the user (processing block 215).

In one embodiment, upon receipt of the request for the experience from the client, the NSP broker sets aside ("carves out" or "secures") bandwidth for the user (including instances where no additional bandwidth is required) to ensure that if many users begin sessions over that network link, over subscription of that link will not cause degradation of the experience for that user. When securing bandwidth, the NSP broker sets aside bandwidth by keeping track of available bandwidth in a database and asking the OSS software to provision more bandwidth if available, when appropriate.

Figure 3 is a flow diagram of one embodiment of a process for obtaining high performance with an end user client built into an application with a performance level option. The application performs many of the operations, but the process of Figure 3 may be performed by processing logic comprising hardware, software, or a combination of both.

Referring to Figure 3, the process begins by processing logic receiving a log in request from the end user (processing block 301). The logging in process starts the end user's application session and may include logging into an NSP's or ASP's website. An application starts in response to an end user request (processing block 302). The application may be a content delivery application, content receiving application or a transaction-based application.

The application provides the user an option for enhanced application performance (processing block 303). In one embodiment, the application provides the user option through

a graphical button or prompt. The application may generate a question the user inquiring as to whether the user wants the enhanced performance. In alternative embodiment, the user is provided with the option for an enhanced experience using an offer with an associated price.

The application tests whether the option was accepted (processing block 304). If the offer was not accepted, application performs at an unmodified performance level (processing block 305). If the option is accepted, the end user client initiates a request for an enhanced performance (processing block 306). In one embodiment, the request is made to an NSP broker first. The client could have contacted the ASP Broker first. Then the ASP Broker would contact the NSP Broker. This is just a reversal of which Broker is contacted by the client. In one embodiment, such a broker is the master broker that coordinates with the other brokers to satisfy the request.

Afterwards, the application tests whether enhanced performance is available (processing block 307). If enhanced performance is not available, the application allows each user to use the application in an unmodified performance level (processing block 308). On the other hand, if enhanced performance is available, the application allows the end user to use the application in an enhanced performance level (processing block 309).

In another embodiment, not shown in Figure 3, the test for the availability of the experience is made prior to presentation of the option to the end user.

Figure 4 is a flow diagram of one embodiment of a process for obtaining higher performance with an end user client built into an application with a higher performance level also built in to the application. Operations depicted in Figure 4 may be performed by processing logic and may comprise hardware, software or a combination of both.

Referring to Figure 4, processing begins when a user logs on to start the BVE session (processing block 401). Then an application starts in response to an end user request (processing block 402). Processing logic in the system starts the application. Thereafter, the application launches a user client (processing block 403). In one embodiment, the application launches a user client without intervention by the end user. Once launched, the user client asks for a bandwidth varied experience (an BVE) (processing block 404).

Next, a broker (an NSP and ASP broker) determines whether the BVE is available (processing block 405). If the BVE is not available, the same broker provides options for the user (processing block 406). If the BVE is available, the BVE is provided to the end user (processing block 408).

One or more techniques are described below to effectuate collaboration between NSPs, ASPs, and end user hosted content sources. In so doing, the likelihood that end users will enjoy applications usage and content downloads with bandwidth and performance characteristics at optimal levels for those experiences may be increased.

In one embodiment, the end user downloads a client in the form of a Java applications player from the Internet. Certain applications or content downloads can be initiated by pointing and clicking on a launch button. The end user selects an experience by clicking on the icon representing that experience and launches the experience by pointing and clicking on the launch button.

The user client notifies an NSP applications broker selected by the end user, and the NSP broker verifies the availability of the requisite bandwidth. The user client also notifies the ASP broker of the request for the application or content, and assuming the NSP broker confirms bandwidth, the NSP triggers the application's session or content download. During this time, the NSP broker secures the requisite bandwidth and service characteristics for the end user by interfacing directly to the network elements, to network element device manager software, or to the NSP's middleware used to manipulate network elements. Such middleware may include what are known in the art as provisioning software, activation software, and policy servers.

Figure 5 is a flow diagram of one embodiment of a process for an NSP broker to determine the readiness for the bandwidth varied experience (BVE) from, for example, NSP elements, middleware and/or caches. Such caches may be at the client, an NSP, or at a content location. Operations in the processing are performed by processing logic in the NSP or ASP broker that may comprise hardware, software, or a combination of both.

Referring to Figure 5, the NSP broker or ASP broker receives a request from an end user client (processing block 501). Next, the NSP broker queries middleware or network elements for network availability (processing block 502). In the alternative, not depicted, the ASP broker initiates determination of BVE availability. The elements may be queried directly, using an element's device manager; through middleware (e.g., server software in NSP) activation or provisioning, or by creating a cache at an NSP.

Next, the NSP broker determines whether adequate performance is available (processing block 503). In one embodiment, the NSP broker determines the availability of adequate performance by querying the middleware or the devices at the NSP. In one embodiment, if the BVE is not available, the NSP broker generates a report to the end user

and/or provides the end user with options (processing block 504). Optionally, if the elements are not available, the NSP broker reports the information to the cache operating system for learning purposes and the caching routines may be adjusted to enable greater BVE availability.

Next the user determines whether to use the modified options and try again (processing block 505). If not, the user receives the regular content experience (processing block 506).

If the bandwidth is available or if the user decides the use a modified option and try again, the NSP broker secures the requisite bandwidth and sends a request to the ASP broker to determine if sufficient resources are available (processing block 507). In response thereto, the ASP broker queries the content storage cache (processing block 508).

The ASP determines whether content storage or cache being used or is available (processing block 509). If content storage or cache is available, the ASP updates the cache statistics (processing block 510). If the content is not available or after updating cache statistics, a broker tests whether upgraded context is available (processing block 511). If not, the process transitions to processing block 506 where the user receives a regular content experience. If content is available, the broker provides the end user with an upgraded content experience (processing block 512).

Optionally, if a content cache is not available, broker generates a report to the cache operating system for learning purposes and the caching routines are modified (e.g., change the LRU routine) for greater availability, such as by changing routines that create space in the cache. Note that caches are not required and need not be employed to provide an end user with a BVE.

The end user interacts with the application or views the content until she/he terminates the session or the session concludes upon reaching its end point. At this time, the NSP broker removes the connection and returns the end user's connection to its default bandwidth setting. The NSP broker obtains usage information and provides that information to the end user client or to the ASPs billing system for generation of a bill. The usage information may be obtained from the involved facilities, or from a device manager, or from billing middleware. The bill may be generated to the end user or to the merchant or advertiser who has agreed to pay for the end user's experience.

The brokers may reside at end-user sites in the case of one or more end-users whose hosts are within enterprise LANs.

The NSP and the ASP may be the same entity, in which case there is only an end user client and one broker. In this embodiment, the same broker is serving the roles of NSP broker and ASP broker.

In one embodiment, there are multiple NSPs that form the network path between the end user location and the ASP. In this case, there is an end user client, NSP brokers, and one ASP broker.

User Client Proxy

In another embodiment, the end user client has an auxiliary extension, the user client proxy (Proxy), that is loaded by the host operating system at boot time. In one embodiment, the client proxy is written in C++ and is loaded under Windows 95/98/NT/2000, MacOS 9.x, or Palm. In an alternative embodiment, the client proxy is launched as a daemon, such as, for example, under the UNIX-based operating systems Linux, MacOS 10.x, and Solaris.

The Proxy serves to monitor network traffic to determine whether the application is receiving the service level it has requested. The Proxy may also serve to proxy requests for those applications which are not launched through the application player interface. The Proxy monitors when any application uses the network, and if it matches a predetermined set of network-usage characteristics, the Proxy launches the applications player to allow the user to acknowledge whether modified service is desired for the application. Finally, the Proxy serves to perform a traceroute or other path determination algorithm to determine which NSPs are in the path between the end-user's application and the content source, at the network layer.

Figure 6 is a flow diagram of a process illustrating operation of the end user client proxy performing traffic monitoring. The processing is performed by processing logic in the client proxy that may comprise hardware, software, or a combination of both.

Referring to Figure 6, the process comprises the proxy receiving requests for an end user client (processing block 601). The proxy then monitors the network traffic conditions (processing block 602). The proxy compares the traffic conditions to the application's requirements (processing block 603). Then the proxy determines whether there is enough bandwidth for the current application's requirements (processing block 604). If there is not, the proxy prevents applications at enhanced levels from being initiated (processing block 605). If they are adequate, the user is given either an option to initiate the BVE or a launcher that may be used to launch the BVE (processing block 606).

Figure 7 is a flow diagram of a process illustrating operation of end user client proxy to determine the identity of the NSPs required to provide the BVE and availability of application tone on behalf of the client. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 7, the process comprises the proxy receiving requests from an end user client (processing block 701). The proxy then determines the identity of NSPs between an end user and content (processing block 702). In one embodiment, a trace route or other path determination method may be used. The proxy determines application tone (processing block 703). Then the proxy determines whether the application tone available across NSP jurisdictions (processing block 704). If not, the proxy does not initiate the BVE (processing block 705). If the application tone is available, the proxy initiates the application or enables enhanced performance options (processing block 706).

In yet another embodiment, common code shared by the application player and the Proxy is stored in a library that resides in the host operating system's shared library directory. This library contains routines for formulating experience requests and manipulating related data structures used by software that orchestrates adding services. In one embodiment, the library is available for use by end-user applications to enable them to request enhanced service without the user going through an applications player and without the Proxy having to proxy a request by monitoring the application's activity on the network. An API may be provided for application developers to take advantage of the library.

In another embodiment, the proxy is configured to launch the applications player when it detects that the user has launched an application that can benefit from enhanced service. In the proxy may be designed to launch the web page containing the application player Java applet or to launch the local native applications player.

Figure 8 is a flow diagram of a process illustrating an application using such a library to invoke enhanced service. The process is performed by processing logic invoked with the application that may comprise hardware, software, or a combination of both.

Referring to Figure 8, the process comprises the user initiating the application (processing block 801). The application then uses the library routines to phrase application performance requirements in symbology that is understandable by the Broker (processing block 802). The routines are initiated by the application (processing block 803). The application then determines whether adequate service is available (processing block 804) by contacting the Broker. If not, the application advises and presents the user with options

(processing block 805) If there is adequate service available, the application causes enhanced performance to be initiated and the user begins use of the application with enhanced performance (processing block 806).

In another embodiment, the application player library contains routines for creating and manipulating relevant data structures. These structures may include: the Profile, which describes the capabilities of network elements and the requirements of applications; the Account, which describes an end-user in the application tone system; the Service, which describes a service offering by an NSP or ASP; the Transaction, which describes a single instantiation of a service offering; and the Message, used in interprocess communication between brokers or between a broker and a client. The library may be used by the applications player and by the applications player client and is also available for use by end-user applications that write to the applications player API.

In one embodiment, the Profile is implemented in XML to encourage third parties to write Profiles that describe their applications and network elements.

The Applications Player

In one embodiment, the applications player is implemented as a Java applet. The Java applet may be invoked from a web page or as a native program from a control strip, a toolbar or an icon-based mechanism on the client host's desktop. When invoked, the applications layer brings up a window (e.g., approximately 120 pixels tall by 200 pixels wide).

Figure 9 illustrates a desktop view of one embodiment of the applications player's window. In one embodiment, the window provides an interface resembling a cell phone or palmtop computer. Referring to Figure 9, the applications player 901 and the web page 902 are shown.

Figure 10 illustrates the launch pane of one embodiment of the applications player. Referring to Figure 10 at the bottom of the window, is an elongated button marked *My (user's vendor's homepage)* 1001. Directly above this is a set of five buttons in a row: *Launch* 1002, *Enable* 1003, *Prefs* 1004, *Log* 1005, and *Help* 1006. Above these buttons is a small text box 1007 displaying the name of an NSP or ASP. On either or both sides of the text box is an arrow (e.g., arrows 1008 and 1009), allowing the user to scroll through a selection of service providers. Together, these buttons 1001-1006 and the text box 1007 comprise the lower 1/3 of the application player window. The upper 2/3 consists of a display area that can display one of five panes, depending on which of the five buttons is selected below. In one

embodiment, at the very top of the applications player window is the applications player vendor logo. It would be apparent to those skilled in the art that the interface may provide access to other features.

In one embodiment, a pull tab on the right side of the applications player extends a calendar page to the right. The calendar is approximately the same height and width as the applications player itself.

In one embodiment, a pull tab 1010 on the bottom of the applications player causes it to collapse so that only the lower 1/3 is displayed to save desktop real estate. In one embodiment, the applications player cannot be collapsed while the calendar is displayed. Figure 11 illustrates one embodiment of a collapsed view of the applications player.

Figure 12 illustrates another pane of the applications player. Referring to Figure 12, an enable button 1201, when checked, causes an application that is already running to receive improved service. In one embodiment, only one application at a time may receive improved service.

Figure 13 illustrates launching an application from the application player. As shown, both the launched applications and the application player remain on the display. Figure 14 illustrates one embodiment of a summary pane that sets forth a log of transactions related to the BVE.

In another embodiment, the applications player Java applet is invoked when the user visits a web page containing the applet. This occurs at the user's local NSP, an ASP, an application player vendor's or other predetermined ".com", ".org", or ".net" (etc.) web site. In either of the former two cases, the service provider is an application tone-enabled provider. In the latter case, the provider enabled to receive services dynamically at run time. In one embodiment, there is no actual content hosted at the application player's web site. Figure 15 illustrates an exemplary web page.

The applications player may be implemented in C++ as a browser plug-in. Alternatively, the applications player may be implemented as a standalone application for any or all of the standard PC and Wintel platforms as well as Linux, Solaris, MacOS and Palm.

Figure 16 is a flow diagram of a process illustrating a user initiating an experience at a service provider's web site. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 16, the process comprises the user launching the browser (processing block 1601). The user then visits the provider's web site (processing block

1602). The web site may be hosted at the NSP, ASP or as a NSP user-tailored page (e.g., my.provider.com). The user is presented with a list of available experiences (processing block 1603). The user selects and launches BVE (processing block 1604). The service provider initiates a request through a surrogate user client (processing block 1605). The request is then initiated using NSP and ASP brokers (processing block 1606).

In another embodiment, Enterprise Java Beans (EJB) technology is used for handling the transactions that occur between the applications player and the local NSP broker.

An Exemplary Broker

In one embodiment, the broker is implemented as a set of daemons running on a Solaris/Sparc platform with configuration being performed using text files (for bootstrap configuration) and HTTP (for runtime configuration). Brokers run at NSPs and ASPs. Brokers may also run at enterprise customer locations.

In one embodiment, the user visits an ASP's web site with the purpose of opening an application tone account. The user enters demographic information and billing information directly into a web page via a common gateway interface (CGI) form interface. The user is prompted for an email address, which becomes the user's enhanced applications username. The user is asked whether he or she is connecting from a host connected to the user's home NSP. If so, the user's home NSP is determined by looking at the IP address of the user's host. This information is available via CGI variables. If not, the user is prompted to identify their home NSP. All of this information is stored by the broker at the home NSP and is also forwarded to the registry by the broker at that NSP.

Figure 17 is a flow diagram of one embodiment of a process by which an end user learns of available experiences by registering at ASP's web site. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 17, the process comprises the user initiating the browser (processing block 1701). The user then registers/signs in at ASP's web site (processing block 1702). The ASP matches the user's registration info/info in a cookie to an available experience list (processing block 1703). The ASP matches experiences for which user qualifies to those made available by user's NSP (processing block 1704). Based on this matching, the user is presented with a list of available experiences for which the user qualifies (processing block 1705). The user selects an experience (via, e.g., point and click) (processing block 1706). In response to the selection, the ASP initiates experience on behalf

of the user or allows the user client to download content or an application for local use (processing block 1707).

In another embodiment, the user visits a central registry and performs the same process. The central registry acts as a clearing house for end users seeking application tone enabled experiences and for value chain participants who are motivated to contribute their hardware, software, and services towards making end user initiated variable-bandwidth experiences available. Such value chain participants may include service providers, content providers, middleware software vendors, hardware vendors, and advertisers. In this case, the broker receiving the information is itself the central registry.

Figure 18 is a flow diagram of one embodiment of a process by which an end user learns of available NSPs by registering at E-Hub, which is a third party web site provided for this purpose. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 18, the process comprises the user initiating the browser (processing block 1801). Then the user signs-in/registers at E-Hub (processing block 1802). Based on registration information or cookie information, the E-Hub matches the user profile to available experiences (processing logic 1803). The matching may not be based on any qualifications. However, qualifications may be based on zip code, area code, etc., or based on answers to questions.

The E-Hub presents user with a list of available experiences for which the user is qualified (processing block 1804). From the list, the user selects experiences which appeal to the user (processing block 1805). The E-Hub matches the user-preferred experiences to a database of NSPs enabling experiences (processing block 1806). In one embodiment, the E-Hub maintains an independent database that is the same database used in the Registry discussed below. The E-Hub provides the user with a ranked list of NSPs affording experiences preferred by the user (processing block 1807). Note that providing a ranked list is not necessary.

In another embodiment, the user visits his or her own home NSP's web site and performs the same process, without having to identify the home NSP (since it is the NSP whose web site is collecting the information).

In one embodiment, at the end of the registration process, the user has an account that is valid at any application tone-enabled NSP and can be used to purchase content hosted at any application tone-enabled ASP.

In one embodiment, the user invokes the applications player locally or from the local NSP's web site or from an ASP's web site. The user selects a content source from the list of available content. The applications player instructs the Proxy to contact the ASP broker which brokers the content. In one embodiment, the Proxy runs a proactive test between itself and the ASP broker. In one embodiment, the Proxy tests throughput, delay, and jitter. The results of the test are displayed on the applications player as a graphical display (e.g., a bar graph, histogram, color temperature diagram, etc.).

In another embodiment, the user invokes the applications player locally or from the local NSP's web site or from an ASP's web site. The user may have reached this page via a bookmark that was added when the user registered with the NSP (if it is the home NSP). Alternatively, the user may have reached this page via a link from a directory web page maintained at application tone vendor's or other web site. The application player displays a list of available content at ASPs enabled to provide added services that can be reached via this local NSP either directly, or through other NSPs enabled to provide added services.

The user selects a content source from the list of available content and requests a demonstration of improved content delivery. In one embodiment, the applications player sends the applications player card number to the ASP broker. In another embodiment, the end user client provides identification information to the ASP without the use of an applications player or player card. The ASP broker initiates a request for end-to-end service and brokers the arrangement with all of the affected NSP brokers. If this process is successful, a sample of the improved service is sent using the requested content.

Figure 19 is a flow diagram of one embodiment of a process by which an end user obtains a sample of enhanced service. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 19, the process comprises the user invoking a client (processing block 1901). The client may be invoked from a user site. Alternatively, the client may be invoked from a provider site. The user is then given an opportunity to view a sample BV experience (processing block 1902). Afterwards, the user invokes an option to view the sample (processing block 1903) and notifies the ASP broker that the end user wishes to view the sample (processing block 1904). The ASP broker negotiates delivery from an ASP source through an NSP-brokered jurisdiction (processing block 1905) and then the user experiences a sample (processing block 1906).

In one embodiment, the list of content is displayed as icons or as some other identifiable indicia. If the content are media, the specific icons used may be the icons for the applications that are launched to view that type of content. If the content comprises remote applications, the icons for the remote applications themselves may be displayed. For each icon, a bar graph or other visual indicia is displayed on a side of the icon (e.g., to the right) indicating the application tone quality that is available for that content through the network connection.

The user selects an icon for desired content. If any additional billing information is needed for the selected content, such as an application player card number or other identification information, the user is prompted to enter it. This prompt occurs if the content is hosted on an ASP with which the user does not already have a service contract. In one embodiment, if scheduling is desired, the user is given a calendar interface to schedule the service.

If the user is to choose between available service levels for this content, the user is prompted for this choice. The choice may be presented in easy to understand terminology, such as Silver/Gold/Platinum, First Class/Coach, Front-row Seats/Balcony. After a selection has been made, the local NSP broker determines if sufficient network resources are available for the content at the level of service that the user has requested (and desires to pay for). If resources are not available, the user is offered a choice of a lower service level, if this is possible. If it is not possible, the user is informed that the desired content cannot be obtained at this time. Note that these may comprise the options discussed herein when a BVE is not available to the end user.

If a service level is available and has been agreed upon, the local NSP broker performs the process to activate the network connection from the local NSP all the way to the content source, going through other NSPs if necessary. If the content is a media file or streaming audio or video, the application for that content may be launched from the user's hard drive and the user is given the URL for the content. If the content is a remote application, the remote application is loaded over the network.

Figure 20 is a flow diagram of one embodiment of a process by which a user selects a content quality level. The processing is performed by processing logic that may comprise hardware, software, or a combination of both.

Referring to Figure 20, the process comprises the user registering/signing on to ASP sites (processing block 2001). After registering, the user reviews content selections

(processing block 2002) and performs a point and click operation on their selection (processing block 2003). The user is then prompted for service level selection (processing block 2004). The prompt may be in metaphorical terms. For example, the prompt may indicate front row/balcony, silver/gold, first class/coach. Using the selection, the user then selects a quality level (processing block 2005) and receives the experience at selected level (processing block 2006).

The service ends when the content is finished loading or when the user's billing amount has run out, whichever is applicable to this instantiation of the service. At this time, the brokers remove their associated network reservations, and a final bill for the service is generated by the ASP. This bill is sent to the user or to the sponsor of the experience by the ASP or by the user's home NSP.

In another embodiment, the user launches a network application on his or her client host. The application may start to retrieve remote content from an ASP as it normally would (e.g., starts to download and display a movie trailer) or it may wait until the desired service level is ready. The application may use the library to instruct the Proxy to request a service matching the Profile of the application and the desired content at the ASP. The Proxy then requests a service from the local NSP, which brokers a connection through any intermediary NSP brokers to the ASP broker. If the service cannot be set up, the Proxy returns an error code to the application indicating the cause of failure. If the service can be set up, the application is notified that the service is available and its cost via callback routines in the library. It is the responsibility of the application to either start using the service or to ask the user if he or she wishes to go ahead with the service at the indicated cost. The payment may be pre-authorized by a merchant, advertiser, or employer.

In another embodiment, the Proxy may launch the application player to ask the user if she/he would prefer enhanced service levels. If the offered service level is desired, the application then directs the Proxy to bring up the service.

In each of the last two described embodiments, once the application has finished using the service, the application instructs the Proxy to end the service. The Proxy, in turn, instructs the local NSP broker to terminate service. If the billing period has run out (e.g., with an application player card), the service automatically expires on all of the brokers. In either case, the brokers then remove all of the associated network reservations, and a final bill for the service is generated by the ASP (or NSP). This bill is sent to the user or to the experience sponsor by the ASP or by the user's home NSP.

In another embodiment, the Proxy continuously watches network traffic generated by, and received by, the client host, to determine when an application is trying to access content that may benefit from enhanced service. If it detects an application trying to use the network in a fashion that matches one of a set of profiles predefined for this purpose, it initiates a service request to the local NSP broker. The local NSP broker brokers a connection through any intermediary NSP brokers to the ASP broker. If the service cannot be set up, nothing further happens. If the service can be set up, the local NSP broker indicates what the cost will be to the Proxy. At this point, the Proxy either instructs the local NSP broker to go ahead and set up the service, or the Proxy launches the applications player application to ask the user if it should set up the service at the indicated cost. Or, payment may be pre-authorized by an advertiser, merchant or employer. If the offered service level is desired, the application then informs the Proxy to bring up the service.

Figure 21 is a flow diagram of one embodiment of a process by which a proxy automatically initiates an option for higher service. The operations in the process are performed by processing logic in the client, NSP and/or ASP may comprise hardware, software, or a combination of both.

Referring to Figure 21, the process comprises the user initiating an application or a content distribution (processing block 2101). The Proxy determines that there is available bandwidth for an enhanced experience (processing block 2102). In one embodiment, the determination by the Proxy occurs by the Proxy monitoring traffic levels and "notices" that bandwidth is available for an enhanced experience by comparing monitored traffic level to profiled information for application.

Upon determining availability of an enhanced experience, the proxy initiates a request for a service option to the NSP broker (processing block 2103). The NSP broker verifies availability of enhanced service (processing block 2104). In one embodiment, the NSP broker performs the verification by examining NSP domain and information from ASP broker. The NSP broker presents the Proxy with one or more options and change in cost associated with implementing the options (processing block 2105). The proxy presents the option(s) to user (processing block 2106).

In another embodiment, payment authorization may be pre-assigned by a merchant, advertiser, or employer, and the BVE initiated automatically.

The Proxy detects when the application has finished using the service by continuing to monitor its network traffic. When the application has finished, the Proxy instructs the local

NSP broker to discontinue the service. The brokers then removes the associated network reservations, and a final bill for the service is generated by the ASP. This bill may be sent to the user, or to the sponsor of the experience by the ASP or by the user's home NSP.

In another embodiment, the Proxy has the capability to generate RSVP (Reservation Protocol for IP) reservation requests on behalf of applications. The Proxy may perform this operation by first determining their requirements by watching the protocols generated by the applications and then asking the broker to match those protocols to the best service quality Profile that fits the application. The broker responds with the appropriate Profile and also informs the Proxy if the network is capable of RSVP from end to end. If it is, the Proxy generates an RSVP request using the parameters in the Profile.

In another embodiment, the broker maintains a database containing all or a subset of the resources in an NSP's or ASP's administrative domain and allocates all or a subset of these resources to be used for provisioning application tone-enabled services.

In another embodiment, a content brokering protocol (CBP) is used for service requests, proxying, and brokering between application tone clients and brokers, and between brokers. The CBP may include data structures describing the characteristics or capabilities of network elements, including active network devices such as, for example, routers and switches, and/or passive network devices such as, for example, hubs and individual LAN, Metropolitan Area Network (MAN), or WAN segments. Alternatively, the CBP may include data structures describing the requirements of an application or content delivery, including, but not limited to, its bandwidth, delay, jitter tolerances, and/or requirement for a certain level of security. The CBP includes data structures describing an instantiation of a network service, including but not limited to its bandwidth, delay, or jitter tolerances, or requirement for a certain level of security. The CBP may include data structures describing an end-user's account with an NSP or ASP, including but not limited to the user's demographic information, billing information, host information, and network connectivity information. In one embodiment, the data structures exchanged by the CBP are specified in XML. The messages used in the CBP may be formatted to those used in HTTP. The messages used in the CBP may be encrypted.

Interaction Between Brokers

A local NSP broker receives a request for enhanced service from a client host. The local NSP broker receives a Profile from the client describing the requirements of the

application, and a complete list of NSPs in the path of the intended service, and the ASP that is intended to source the content. The user of this client host has already set up an application tone account with this NSP or with the ASP.

The local NSP broker uses this Profile to allocate resources within the administrative domain of this NSP. If this allocation process is successful, the local NSP broker passes along the Profile to the ASP broker, if there are no intervening NSPs. If there are intervening NSPs, the local NSP broker passes along the Profile to each NSP broker in the path, as well as to the ASP broker. Each intervening NSP broker performs internal resource allocation and reports its status to the local NSP broker. Likewise, the ASP broker performs its resource allocation and reports its status to the local NSP broker. If all intervening NSP brokers (if any) and the ASP broker report successful resource allocation, the local NSP broker sends a success message back to the client. If any brokers report an unsuccessful reservation, the local NSP broker informs all of the brokers to deallocate any resources that they may have allocated and sends a failure message back to the client. If the enhanced service was requested in immediate mode, the local NSP broker informs all other NSP brokers (if any) and the ASP broker to activate the service. Note that activation may occur automatically if provisioning software at the NSPs and ASP is capable of directing activation to start automatically. If the enhanced service was not requested in immediate mode but was for a scheduled service, activation occurs at the scheduled time. Once the application has finished or the scheduled time has ended, each broker deallocates its resources and the links are deactivated.

In one embodiment, the local NSP broker receives a request for enhanced service from a client host. If the user of this client host has an application tone account with this NSP, the local NSP broker requests the cost of the content from the ASP before starting the resource allocation process. The local NSP broker also requests the transport charges for this service from intervening NSPs, if any, by sending this information along with the Profile. The local NSP broker also adds the transport charges of its own NSP. The combined cost information is passed along to the client and the user is asked for approval. If the user does not approve, the request does not proceed. If the user approves, the reservation process continues. The cost may be preapproved by a user, merchant, advertiser, or employer. In another embodiment, the costs may be pre-assigned by cooperating NSPs/ASPs.

At the end of the service, the final cost of the service is calculated, taking into account any variable charges. The cost of the content and transport are added to the user's or sponsor's billing account at the local NSP. The local NSP then reimburses the intervening

NSPs, if any, and the ASP for their charges. Each NSP and ASP broker keeps track of all such charges for sourced traffic or transiting traffic that enters, travels through, and exits the NSP, and costs reported by other brokers are checked against internal records. All charges are settled amongst the service Providers at set time intervals.

Figure 22 is a flow diagram of one embodiment of a process by which an end user invokes an enhanced service paid for by an advertiser. The processing may be performed by processing logic in the end user's client, NSP and ASP that may comprise hardware, software, or a combination of both.

Referring to Figure 22, the process comprises the client host downloading a limited use application card applet (processing block 2201). The application card applet operates similarly to a prepaid phone card. The end user initiates an enhanced experience through the client (processing block 2202). The NSP broker requests authorization from the ASP hosting the sponsored experience (processing block 2203) and determines whether authorization is granted (processing block 2204). If not, the end user receives a report with options (processing block 2205). If authorization is granted, then the NSP broker determines availability of NSP resources to deliver the sponsored experience (processing block 2206).

In another embodiment, the end user initiates the experience without use of an application card applet. The client may be built in to its application or reside at the NSP or ASP portal.

In another embodiment, the local NSP broker receives a request for an application tone service from a client host. If the user of this client host has an application tone account with the ASP hosting the content or application, the local NSP broker requests the cost of the content from the ASP before starting the resource reservation process. It also requests the transport charges for this service from intervening NSPs, if any, by sending this information along with the Profile. The broker also adds the transport charges of its own NSP. The combined cost information is passed along to the client and the user is asked for approval. If the user does not approve, the request does not proceed. If the user approves, the reservation process continues. Or, authorization may be pre-approved by the user, or sponsor, employer, advertiser, merchant, costs having been pre-assigned by cooperating NSPs/ASPs. At the end of the service, the final cost of the service is calculated, taking into account any variable charges including those based on duration of service, type of application or content, and amount of data transferred. The cost of the content and transport are sent to the ASP, which adds the combined cost to the user's (or sponsor's) billing account at the ASP. The ASP then

reimburses the local NSP and intervening NSP (if any) for their transport charges. Each NSP and ASP broker keeps track of all such charges for sourced traffic (where sourced traffic originates at the NSP or ASP) or transiting traffic, and costs reported by other brokers are checked against internal records. All charges are settled amongst the service providers at set time intervals.

In another embodiment, the local NSP broker receives a request for application tone service from a client host. If the user of this client host has an applications player card, an already recognized ability to pay, or an account exists for sponsorship of this experience, the local NSP broker requests the cost of the content from the ASP before starting the resource reservation process. It also requests the transport charges for this service from intervening NSPs, if any, by sending this information along with the Profile. The broker also adds the transport charges of its own NSP. The combined cost information is checked against the maximum charge allowed by the applications player card by contacting the broker at the service provider which established the account for the card. (In many cases, such as with advertising content, this is the ASP.) If the service provider does not approve, the request does not proceed. If the service provider approves, the reservation process continues. At the end of the service, the final cost of the service is calculated, taking into account any variable charges. The cost of the content and transport are sent to the service provider that established the account for the card. The service provider then reimburses the local NSP, the intervening NSPs (if any), and the ASP for their transport and content charges. Each NSP and ASP broker keeps track of all such charges for sourced traffic or transiting traffic, and the costs reported by other brokers are checked against internal records. All charges may be settled amongst the service providers at set time intervals.

In another embodiment, the ASP brokers are replaced by customer-site-situated brokers that manage the end user's enterprise devices and device manages (e.g., video cameras, codecs, electronic whiteboards, etc.). In this embodiment, the application player client signals the need for enhanced performance to the NSP broker and to the customer applications broker which may, in the case of multi-end user collaboration, be represented in multiple instantiations. In this embodiment, the "bill" may be a departmental usage charge incorporated in the enterprise's internal accounting system for tracking resource use among functional areas.

In another embodiment, roaming is enabled. Roaming refers to a way by which application tone brokers allow an end user to use an NSP other than his or her own home

NSP (for example, if the user is traveling), and still receive services via application tone. This requires that the local NSP be application tone-enabled. Note that the service experience may not remain exactly the same, since the user may have a different type of network connectivity or maximum bandwidth, and will likely have access to a different set of content. However, the user will be able to enjoy the same application tone as a user that is native to that particular local NSP.

Figure 23 is a flow diagram of a process illustrating roaming by the end user. Operations in the process may be performed by processing logic in the client, NSP and ASP that may comprise hardware, software, or a combination of both.

Referring to Figure 23, the process comprises the end user attempting to launch an experience from a client connected to a new NSP (processing block 2301). The new NSP broker consults a database list of cooperating NSPs/ASPs (processing block 2302). The database may be at the Registry described below or at the E-Hub. Then the broker determines whether an enhanced service is available across jurisdictions (processing block 2303). If it is not connecting to provide the enhanced service across the jurisdiction, the new NSP provides a report and/or options to the user (processing block 2304). If there is connectivity, then the NSP broker coordinates BVE set up and delivery with cooperating NSP brokers and ASP broker (processing block 2305).

In another embodiment, the NSP broker uses the minimum and maximum bandwidth tags within the service Profile to determine whether to use CBR or VBR PVCs on an ATM (Asynchronous Transfer Mode) network. The NSP broker requests the service provider's OSS (Operational Support Software) provisioning software to reallocate bandwidth between two endpoints within the NSPs network which conform to the path (within this NSP) between the end-user and the content.

In another embodiment, the NSP broker uses a weighted shortest-path algorithm to determine which links within a service provider's network are to be used to carry traffic for a particular content stream. The bandwidth, delay, and/or jitter characteristics of the service profile may be used as weights in the shortest-path algorithm. Once links have been identified, the broker then requests bandwidth to be provisioned and activated on each of the selected links. To prevent lack of bandwidth starvation, the broker may also request bandwidth to be re-provisioned and activated on other links in the NSP's network as a result of this operation.

In another embodiment, the broker instructs the NSP's OSS software to re-provision and/or re-activate bandwidth in a DSLAM in order to support the requirements of a service traversing a twisted-pair copper line. Figure 24 is a block diagram illustrating the broker requesting OSS software to activate resources in the network such as, for example, DSLAMs and ATM switches.

Figure 25 is a flow diagram illustrating one embodiment of a process by which an NSP broker effects service through DSLAM. Operations in the process are performed by processing logic in an NSP that may comprise hardware, software, or a combination of both.

Referring to Figure 25, an NSP broker receives a client's request for service to be delivered over twisted-pair copper (processing block 2501). The broker instructs the NSP OSS to reprovision and/or reactivate connection in DSLAM (processing block 2502). The OSS effects reactivation and/or reprovisioning (processing block 2503) and the NSP delivers the enhanced services to user over twisted-pair (processing block 2504).

In another embodiment, the broker instructs an NSP's OSS software to re-provision and/or re-activate bandwidth in order to support the requirements of a service traversing a wireless network. The bandwidth may be re-provisional and/or re-activated in an MMDS or LMDS distribution point.

Figure 26 is a flow diagram of one embodiment by which an NSP broker effects service in a wireless network. Operations in the process are performed by processing logic in an NSP that may comprise hardware, software, or a combination of both.

Referring to Figure 26, an NSP broker receives a service request from a client over a wireless connection (processing block 2601). The NSP broker instructs the NSP's OSS software to reprovision and/or reactivate bandwidth at distribution point (processing block 2602). As discussed above, the distribution point may comprise an MMDS or LMDS distribution point. The OSS software effects the reactivation/reprovisioning of bandwidth at the distribution point (processing block 2603). The NSP broker then transmits the service to the user over the wireless connection (processing block 2604).

In another embodiment, the broker instructs an NSP's OSS software to re-provision and/or re-activate bandwidth in an IP over SONET link in order to support the requirements of a service traversing an optical fiber. Figure 27 is a flow diagram of one embodiment of a process by which a broker negotiates bandwidth in IP over sonet network. Operations in the process are performed by processing logic in an NSP that may comprise hardware, software, or a combination of both.

Referring to Figure 27, an NSP receives an enhanced service request (processing block 2701) and instructs the OSS software to provision and/or reactivate bandwidth in IP over SONET link (processing block 2702). The NSP OSS provisions/reactivates bandwidth at an add-drop multiplexer or other SONET network element (processing block 2703), and the NSP transmits the service through IP over SONET link (processing block 2704).

In another embodiment, the broker instructs an NSP's OSS software to re-provision and/or re-activate bandwidth in an analog or digital cable distribution point in order to support the requirements of a service traversing a coaxial cable line. Figure 28 is a flow diagram of one embodiment of a process by which an NSP broker negotiates bandwidth in analogue or digital cable network. Operations in the process are performed by processing logic in the NSP that may comprise hardware, software, or a combination of both.

Referring to Figure 28, an NSP broker receives a BVE request (processing block 2801). The request may be for an enhanced service request from a client over a coaxial cable network. The NSP broker instructs OSS software to provision and/or reactivate bandwidth at a cable distributing point (processing block 2802). The cable distribution point may be an analog or digital cable distribution point. The OSS software provisions/reactivates bandwidth at the coax cable distribution point (processing block 2803), and the NSP transmits the enhanced experience through cable to the end user (processing block 2804).

In another embodiment, the broker instructs an NSP or ASP's OSS software to set and/or enforce policies for an Ethernet, gigabit Ethernet, or Fast Ethernet switch in order to support the requirements of a service traversing the switch. Figure 29 is a block diagram illustrating the broker requesting the OSS software to activate resources in a network, such as routers and/or Ethernet switches.

Figure 30 is a flow diagram of one embodiment of a process by which a broker negotiates enhanced service transmission through a gigabit ethernet or fast ethernet. Operations in the process are performed by processing logic in the NSP that may comprise hardware, software, or a combination of both.

Referring to Figure 30, an NSP or ASP broker receives an enhanced service request (processing block 3001). This request may come from a client or an ASP broker or another NSP broker. The broker instructs policy software in a policy server. The policy server may be located at the NSP or ASP to enforce policies over Ethernet, gigabit Ethernet, or fast Ethernet facilities (processing block 3002). The OSS activates policies (processing block

3003) and the NSP broker transmits the enhanced service over the ethernet facilities (processing block 3004).

In another embodiment, the broker instructs an application-level multicasting server at an ASP to add or remove an outgoing data stream to a specified internet address.

In another embodiment, activation is performed by existing OSS activation software (e.g., Syndesis) on an ATM network. The OSS provisioning software (e.g., Syndesis, Architel, etc.) is only instructed to set aside the requisite bandwidth on all elements between two endpoints within the NSP's/ASP's network. The broker-managed provisioning software instructs the activation software to tear down existing PVCs which underlie the end-user's network connection and to replace them with CBR or VBR PVCs as specified by the broker. At the end of the service reservation, the broker instructs the provisioning software to restore original PVCs, and the task is again carried out by the activation software.

In another embodiment, for those applications which require a secure network path, VPN technology is used to create a secure virtual network between the content source and the client host.

In another embodiment, the broker receives feedback about the bandwidth requirements of each content transaction and uses this information to set aside system resources on the ASP's content or applications server for that transaction. These resources may include, but are not limited to, CPU time, access to storage, and network buffer space (e.g., for TCP windows).

In another embodiment, a Registry, which may be a central point to which the client queries, determines which ASP brokers are hosting desired content, and where NSP brokers look up other brokers. In one embodiment, the Registry is implemented as an LDAP-accessible directory server using off-the-shelf directory software running on a e.g., Linux/Intel platform. The Registry may be owned and administered by an application tone vendor. Any information entered into the Registry would then likewise be owned by an application tone vendor who is a third party vendor who is not necessarily an NSP or ASP. In one embodiment, one directory server is initially deployed at a hosting facility, with a second for disaster recovery purposes only. Scalability is addressed by deploying multiple LDAP-accessible servers, each of which is responsible for a portion of the directory. A web server hosts the application tone "dot com" web site to service web queries about joining the Registry.

In another embodiment, the E-Hub "dot com" web site is the central point which users and interested parties query to receive information about application tone. In one embodiment, the E-Hub is implemented as a web server running on a Linux/Intel platform and is owned and administered by an application tone vendor. Any information entered into the E-Hub may be likewise owned by the application tone vendor. One web server is initially deployed at a hosting facility leased by an application tone vendor, with a second for disaster recovery purposes only. Scalability is addressed by deploying multiple web servers using content replication (e.g., Inktomi, CacheFlow, etc.).

In another embodiment, content is hosted at the NSP or at an NSP-proximate ASP under a caching arrangement. In this instance, the NSP or ASP broker instructs the caching system (e.g., CacheFlow's caching system) to set aside resources for the application session or content download. Application tone availability is determined from network devices at the NSP and from statistical probabilities built into the caching device at the NSP.

Alternatives

In one embodiment, the ASP/NSP brokers use third-party products to set up and enforce policies on IP networks. These products, known collectively as policy managers, use standard models (such as Differentiated services, or DiffServ) and/or proprietary models for IP policies. Policy managers are typically implemented as a policy server, and a user interface for accessing that server. The policy server itself is the only part of the third-party products that is used by the ASP/NSP brokers.

In another embodiment, access to the policy servers by the brokers is standardized to use the Lightweight Directory Access Protocol version 3 (LDAP) and Common Open Policy Server (COPS). The architectural and platform implementation of the policy manager is a matter of choice so as long as the policy server itself is accessible via standard protocols. The policy managers use SNMP, HTTP, command-line commands, and other methods for carrying out activation of policies after they are stored in the policy server. Application tone Profile information is converted to the format appropriate to each supported vendor's product.

A RealNetworks server may be enhanced with application tone clients and brokers to enable enhanced downloads and to expand the utility of RealNetworks applications. A Quicktime server may be enhanced with application tone clients and brokers to enable enhanced downloads and to expand the utility of Quicktime applications. A Microsoft Media Player server may be enhanced with application tone clients and brokers to enable enhanced

downloads and to expand the utility of Microsoft Media Player applications. A remote storage server may be enhanced with application tone clients and brokers to enable enhanced network disk activity such as backups, restores, and remote file access. An MP3 or similar music server may be enhanced with application tone clients and brokers to enable enhanced downloads and to expand the utility of MP3 applications. In each of these cases, control of the server is accomplished via server management software that receives and carries out requests by the broker.

Application or content servers may be enhanced with application tone clients and brokers to enable applications or content to be received by end-users on a pay-per-view, pay-per-use, or pay-per-download basis. Billing information is mediated by the application tone brokers to the final billing application to allow the charge for the content to appear on the end-user's NSP bill or on a separate bill.

The embodiments described herein allow for end users to invoke performance-variable experiences situationally and be charged, or have merchants and/or advertisers/employers be charged, only for the incremental usage or value invoked; for ASPs and NSPs to collaborate on new offerings secure in the knowledge that they can provide those offerings to end users at an affordable cost; for end users to invoke bandwidth-variable experiences without delays, errors, and logistical frustrations attendant on labor-centric methods of varying bandwidth; for providers to make such experiences available without having to choose between low margins and high prices; for end users to be required to purchase semi-permanent connections at higher bandwidth; for users without any special technical knowledge to reasonably anticipate whether or not bandwidth available for particular services is a good match for those services; for end users to avoid overpayments for experiences requiring lower performance characteristics; for end users without technical expertise to ascertain whether or not required performance levels are in fact being provided; for allowing end users to decide whether or not to launch experiences on the basis on informed consent, taking into account cost/value considerations; for ASPs and NSPs to determine in a cost-efficient and effective manner end users', employers', and sponsors' interest in and willingness to pay variable-performance experiences; for end users to be able to effect bandwidth-variable experiences involving multiple network jurisdictions; for members of the value chain who wish to participate in providing bandwidth-variable experiences to discover and collaborate with one another; for limitations in regional hosting to be overcome; for limitations in caching solutions to be

overcome; for progress in all these areas to be made incrementally, without requirement that all inhibiting factors be overcome at once.

Thus, there is described herein one or more embodiments of a commercially viable end user driven commerce system (or systems) that contains the needed features outlined above and which addresses the above-described shortcomings in the prior art.

Therefore, in at least one embodiment, end users are able to invoke experiences—interactions with applications and content downloads—which require variable bandwidth and for which the end users are charged based on actual usage. Also, employers, merchants and advertisers are able to make such experiences available to end users and be charged for the end users' experiences on the basis of actual usage.

In at least one embodiment, end users can access new applications and content that have substantially greater or more consistently reliable performance characteristics, and to pay or have sponsors pay for such experiences without permanently increasing the bandwidth afforded by network service providers. Moreover, applications and content developers can exercise creativity in the development of new experiences with the assurance that end user access for these experiences will be affordable, while application and network/hosting providers to make such experiences available with the knowledge that end users will actually be able to purchase them at reasonable costs.

In one embodiment, end users can invoke experiences without encountering the delays, opportunities for error, and frustration caused by the current state of the art, and can do so affordably. Providers can make these experiences available without the requirement of substantial labor costs and the forced trade-off between high prices and low margins, and end users and providers can transact for variable-performance experiences without the need of creating “nailed up” semi-permanent connections with higher bandwidth.

End users are able to determine if and to what extent network providers have enabled connections for different applications and content downloads, and can then, as with mobile phone service, make informed decisions about whether or not to invoke a particular experience.

A mechanism for providing informed consent to the end user is also described. With such a notification service she/he can determine if the extra cost for an experience is worthwhile, or if someone else is willing to pay for it. She/he can also avoid overpaying for an experience in instances where the bandwidth/quality characteristics of her/his existing connection are adequate.

End users can readily ascertain if the quality of service for an experience is a good match to the requirements of that experience. The mechanism described herein enable end users with no appreciable technical skills to make that determination. Moreover, distribution and hosting providers can have an automated mechanism of ascertaining the end user's interest in modifying bandwidth/quality characteristics and making those changes automatically. The increase in traffic and hosted content can then provide a greater return on the infrastructure investment.

End users, network service providers, and application/content providers can have a highly-automated mechanism of effecting bandwidth changes in an automated and collaborative manner, so that the providers are incented to create new experiences for end-users, and so that both applications/content as well as network/hosting providers can generate new sources of revenue from advertisers, merchants, employers, and end users.

On-demand end-user initiated bandwidth-variable experiences are provided that can be delivered in an automated mechanism across network jurisdictions. In one embodiment, such a solution includes the ability to cope with different environments and protection of the network assets from unwarranted intrusion by others.

A clearinghouse type operation that, on an automated basis, provides information to participants in the value chain regarding need for and willingness to pay for new or existing services, and members of the value chain are able to identify one another in order to foster cooperation among themselves towards the development and enhancement of services.

As a higher level embodiment, a system of related solutions that can address the deficiencies in the current art in a modular fashion, improving end users' success with applications and content delivery incrementally. Each deficiency in the current art inhibiting both end user success with experiences and pricing improvements should be cured on an as-you-go basis, so that substantial improvements are made before a comprehensive solution set is attainable.

An Exemplary Network

Figure 31A is a block diagram of one embodiment of a network environment 3101 that may be used in the translation technique described above. In one embodiment, a server computer system 3100 is coupled to a wide-area network 3110. Wide-area network 3110 may include the Internet or other proprietary networks including, but not limited to, America On-

Line™, CompuServe™, Microsoft Network™, and Prodigy™. Wide-area network 3110 may include conventional network backbones, long-haul telephone lines, Internet and/or Intranet service providers, various levels of network routers, and other conventional mechanisms for routing data between computers. Using network protocols, server 3100 may communicate through wide-area network 3110 to client computer systems 3120, 3130, 3140, which are possibly connected through wide-area network 3110 in various ways or directly connected to server 3100. For example, client 3140 is connected directly to wide-area network 3110 through direct or dial-up telephone or other network transmission line.

Alternatively, clients 3130 may be connected through wide-area network 3110 using a modem pool 3114. Modem pool 3114 allows multiple client systems to connect with a smaller set of modems in modem pool 3114 for connection through wide-area network 3110. Clients 3131 may also be connected directly to server 3100 or be coupled to server through modem 3115. In another alternative network typology, wide-area network 3110 is connected to a gateway computer 3112. Gateway computer 3112 is used to route data to clients 3120 through a local area network 3116. In this manner, clients 3120 can communicate with each other through local area network (LAN) 3116 or with server 3100 through gateway 3112 and wide-area network 3110. Alternatively, LAN 3117 may be directly connected to server 3100 and clients 3121 may be connected through LAN 3117.

Using one of a variety of network connection mechanisms, server computer 3100 can communicate with client computers 3150. In one embodiment, a server computer 3100 may operate as a web server if the World-Wide Web ("WWW") portion of the Internet is used for wide area network 3110. Using the HTTP protocol and the HTML coding language, or XML, such a web server may communicate across the World-Wide Web with a client. In this configuration, the client uses a client application program known as a web browser such as the Netscape™ Navigator™, the Internet Explorer™, the user interface of America On-Line™, or the web browser or HTML translator of any other conventional supplier. Using such browsers and the World Wide Web, clients 3150 may access graphical and textual data or video, audio, or tactile data provided by the web server 3100.

Figure 31B illustrates an end to end arrangement by which end users access content from content servers through various elements where access to the content is controlled by an ASP broker and access to portions of the network are controlled by two separate NSP brokers.

Figure 31C illustrates multiple physical paths by which an end user gains access to the network via a network element having an associated NSP broker. The access to the network results in multiple tasks being provided to a network element that has an associated ASP broker that controls access to content servers.

Figure 31D illustrates multiple client collaboration in which end users gain access to the network through a network element with an associated consumer broker (a broker with the same functionality as an NSP broker, but which resides within a customer's environment) or NSP broker.

Figure 31E illustrates an inter-provider arrangement by which two service provider boundaries are shown connected to each other via network elements. Each service provider has an associated NSP broker or ASP broker, by which they arrange transactions for network resources in bulk.

Figure 31F illustrates a multi-client multi-provider arrangement in which end users access a network through the same service provider using a network element with an associated consumer broker. This service provider has an associated NSP broker. Network elements in one service provider are used to transfer information between the service providers, with at least one of the service providers having an NSP broker associated with the network elements.

An Exemplary Computer System

Figure 32 is a block diagram of an exemplary computer system. Referring to Figure 32, computer system 3200 may comprise an exemplary client 150 or server 100 computer system. Computer system 3200 comprises a communication mechanism or bus 3211 for communicating information, and a processor 3212 coupled with bus 3211 for processing information. Processor 3212 includes a microprocessor, but is not limited to a microprocessor, such as, for example, Pentium™, PowerPC™, Alpha™, etc.

System 3200 further comprises a random access memory (RAM), or other dynamic storage device 3204 (referred to as main memory) coupled to bus 3211 for storing information and instructions to be executed by processor 3212. Main memory 3204 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 3212.

Computer system 3200 also comprises a read only memory (ROM) and/or other static storage device 3206 coupled to bus 3211 for storing static information and instructions for processor 3212, and a data storage device 3207, such as a magnetic disk or optical disk and its corresponding disk drive. Data storage device 3207 is coupled to bus 3211 for storing information and instructions.

Computer system 3200 may further be coupled to a display device 3221, such as a cathode ray tube (CRT) or liquid crystal display (LCD), coupled to bus 3211 for displaying information to a computer user. An alphanumeric input device 3222, including alphanumeric and other keys, may also be coupled to bus 3211 for communicating information and command selections to processor 3212. An additional user input device is cursor control 3223, such as a mouse, trackball, trackpad, stylus, or cursor direction keys, coupled to bus 3211 for communicating direction information and command selections to processor 3212, and for controlling cursor movement on display 3221.

Another device that may be coupled to bus 3211 is hard copy device 3224, which may be used for printing instructions, data, or other information on a medium such as paper, film, or similar types of media. Furthermore, a sound recording and playback device, such as a speaker and/or microphone may optionally be coupled to bus 3211 for audio interfacing with computer system 3200. Another device that may be coupled to bus 3211 is a wired/wireless communication capability 3225 to communication to a phone or handheld palm device.

Note that any or all of the components of system 3200 and associated hardware may be used in the present invention. However, it can be appreciated that other configurations of the computer system may include some or all of the devices.

Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims which in themselves recite only those features regarded as essential to the invention.

CLAIMS

We claim:

1. A method comprising:
receiving a request for a bandwidth-varied experience (BVE) from a client; and
a service provider dynamically providing services to satisfy the BVE request.
2. The method defined in Claim 1 further comprising an NSP broker and an ASP broker cooperating to provide the BVE.
3. The method of Claim 1 wherein dynamically providing services comprises the service provider reallocating network resources.
4. The method of Claim 1 wherein dynamically providing services comprises the service provider enabling network resources.
5. A method comprising:
means for receiving a request for a bandwidth-varied experience (BVE) from a client;
and
means for providing services to satisfy the BVE request.
6. The method of Claim 1 wherein the means for dynamically providing services comprises means for reallocating network resources.
7. The method of Claim 1 wherein the means for dynamically providing services comprises means for enabling network resources.
8. A computer software product having one or more recordable medium having executable instructions stored thereon which, when executed by a processing device, cause the processing device to:
receive a request for a bandwidth-varied experience (BVE) from a client; and

dynamically provide services to satisfy the BVE request.

9. A method for providing a bandwidth-varied experience (BVE) in a network, the method comprising:

- a first broker receiving a request for the BVE;
- a network service provider (NSP) broker generating a query to determine if adequate network resources are available to provide the BVE;
- the NSP broker sending a request to an application service provider (ASP) broker for the BVE;
- the ASP broker querying an ASP content source to determine if the BVE is available in response to a request from the NSP broker;
- delivering the BVE if the NSP broker receives an acknowledgment regarding availability of network resources in the network and the ASP broker receives an indication of availability from the ASP content source.

10. The method defined in Claim 9 wherein delivering the BVE comprises the NSP securing requisite bandwidth for the BVE.

11. The method defined in Claim 9 further comprising a client generating a request automatically in response to activation of an application by an end user.

12. The method defined in Claim 9 wherein the BVE comprises a higher bandwidth network connection.

13. The method defined in Claim 9 wherein the BVE comprises an improved connection along with an application from an ASP.

14. The method defined in Claim 9 wherein the NSP broker comprises an NSP with installed brokering software and the ASP broker comprises an ASP with installed brokering software.

15. The method defined in Claim 9 wherein the BVE comprises receiving a lower bandwidth.

16. The method defined in Claim 9 wherein the BVE comprises deferring receipt of a particular service.
17. The method defined in Claim 9 wherein the NSP broker queries middleware or network elements to determine network availability.
18. The method defined in Claim 17 wherein the NSP broker queries elements in at least one of the following ways: directly, using an elements device manager, through middleware activation or provisioning or by querying a cache at the NSP.
19. The method defined in Claim 9 further comprising the NSP broker performing billing transactions to obtain payment for the added service.
20. The method of Claim 9 further comprising:
the NSP broker requesting usage information from billing middleware;
the NSP broker forwarding the usage information to an NSP billing system; and
creating a bill for the BVE.
21. The method defined in Claim 19 wherein the bill is created for an end user client.
22. The method defined in Claim 19 wherein the bill is created for a sponsor.
23. The method defined in Claim 19 wherein the bill is created for an employer.
24. The method defined in Claim 9 wherein the request is from a client network interface.
25. The method defined in Claim 24 wherein the client network interface comprises one from the group of a Digital Subscriber Line (DSL) modem, a cable modem, a wireless modem, or a channel service unit.

26. The method defined in Claim 9 further wherein the first broker receives the request for the BVE from a selection by an end user via an applications player.

27. The method defined in Claim 26 further comprising:
displaying a window representing at least either available network services and applications; and
receiving a selection of one or more of the available network services and applications from an end user.

28. The method defined in Claim 9 further comprising prompting a user to pay for a service in response to the request for the BVE.

29. The method defined in Claim 9 further comprising receiving pre-authorization of payment for the BVE.

30. The method defined in Claim 29 wherein the pre-authorization comes from one of the following groups: sponsor, advertiser, merchant, or employer.

31. The method defined in Claim 30 further comprising negotiating a financial transaction within an end user to obtain compensation for an added network service to provide the BVE.

32. The method defined in Claim 9 wherein delivering the BVE comprises configuring individual network elements to satisfy the service level associated with the BVE.

33. The method defined in Claim 9 further comprising the NSP broker requesting bandwidth from NSP elements.

34. The method defined in Claim 33 wherein requesting bandwidth from NSP elements comprises the NSP broker requesting bandwidth by provisioning or activating NSP elements to obtain requisite bandwidth.

35. The method defined in Claim 34 wherein requesting bandwidth from NSP elements comprises the NSP broker maintaining a predetermined bandwidth at a level of service.

36. The method of Claim 33 further comprising NSP elements acknowledging availability of requisite bandwidth.

37. The method of Claim 33 further comprising the NSP elements reporting a failure of the requests when adequate bandwidth is unavailable.

38. The method of claim 37 further comprising:

The NSP elements reporting a failure of the request when adequate bandwidth is unavailable; and

the NSP broker reporting unavailability of the BVE to an end user client.

39. The method defined in Claim 9 further comprising:

the ASP broker querying a content location to determine if memory capacity is available;

updating memory access statistics if the memory is being used; and

the ASP broker determining if upgraded content is available.

40. The method defined in Claim 9 further comprising:

an application providing an end user an option for enhanced performance in response to an end user starting an application;

determining whether the option was accepted;

an end user client initiating a request for an enhanced performance.

41. The method defined in Claim 9 further comprising:

an end user starting an application;

the application launching the user client; and

user client requesting generating the request for the BVE.

42. The method defined in Claim 9 further comprising:

a proxy of the client receiving the request from an end user;
the proxy monitoring network traffic conditions;
the proxy comparing traffic conditions to application requirements to determine if the application is receiving a requested service level;
the proxy providing an end user with a window by which the end user can select the BVE.

43. The method defined in Claim 9 further comprising:
a proxy of the client monitoring when an application uses the network;
determining whether the application has a predetermined set of network/usage characteristics;
the proxy launching an applications player to allow the user to acknowledge whether a modified service is desired for the application if the predetermined network-usage characteristics exist;
the proxy determining which NSPs are in the path between an application of the end user and the content source.

44. The method defined in Claim 43 further comprising the proxy comparing the traffic conditions to application requirements.

45. The method defined in Claim 9 further comprising:
a proxy of the client receiving the request from an end user for the BVE;
the proxy determining identity of NSPs between the end user and predetermined content;
the proxy determining whether the network is capable of providing the BVE;
the proxy enabling the BVE for the client.

46. The method defined in Claim 45 further comprising determining whether the network is capable of providing the BVE across NSP jurisdictions.

47. The method defined in Claim 9 further comprising:
the NSP broker removing a connection; and
returning a connection of the end user to its default bandwidth setting.

48. An architecture for providing a bandwidth-varied experience (BVE) in a network, the architecture comprising a client host, a network service provider (NSP) coupled to the client host via a network connection, and an application service provider (ASP) coupled to the NSP, wherein

the NSP comprises an NSP broker to request and secure bandwidth from elements in the network to determine if adequate network resources are available to provide the BVE, the NSP broker sending a request to an ASP broker for the BVE,

the ASP broker queries an ASP content source to determine if the BVE is available in response to a request from the NSP broker;

the NSP delivers the BVE if the NSP broker receives an acknowledgment regarding availability of network resources in the network and the ASP broker receives an indication of availability from the ASP content source.

49. The system defined in Claim 45 wherein the client host displays a window representing at least one service or application available to an end user.

50. An apparatus for providing a bandwidth-varied experience (BVE) in a network, the apparatus comprising:

means for receiving a request for the BVE;

means for requesting bandwidth from elements in the network to determine if adequate network resources are available to provide the BVE;

means for sending a request to an application service provider (ASP) broker for the BVE;

means for querying an ASP content source to determine if the BVE is available in response to the request;

means for delivering the BVE if an acknowledgment is received regarding availability of network resources in the network and the ASP broker receives an indication of availability from the ASP content source.

51. The method defined in Claim 10 wherein the means for delivering the BVE comprises means for securing requisite bandwidth for the BVE.

52. The apparatus defined in Claim 50 wherein the BVE comprises higher bandwidth network connection.

53. The apparatus defined in Claim 50 wherein the BVE is a bandwidth network connection that does not drop below a predetermined value.

54. The apparatus defined in Claim 50 further comprising means for performing billing transactions to obtain payment for the added service.

55. The apparatus defined in Claim 50 wherein the means for requesting comprises a client network interface.

56. The apparatus defined in Claim 55 wherein the client network interface comprises one from the group of a Digital Subscriber Line (DSL) modem, a cable modem, a wireless modem, or a channel service unit.

57. The apparatus defined in Claim 50 further comprising:
means for querying a content location to determine if memory capacity is available;
means for updating memory access statistics if the memory is being used; and
means for determining if upgraded content is available.

58. The apparatus defined in Claim 50 further comprising:
means for receiving a request from an end user;
means for monitoring network traffic conditions at a client;
means for comparing traffic conditions to application requirements at the client to determine if the application is receiving a requested service level;
means for providing an end user with a window by which the end user can selected the BVE.

59. The apparatus defined in Claim 50 further comprising:
means for receiving a request from an end user for the BVE;
means for determining identity of NSPs between the end user and predetermined content;

means for determining whether the network is capable of providing the BVE; means for the proxy enabling the BVE for the client.

60. A method comprising:
launching a player;
initiating a request for bandwidth necessary for an experience;
determining whether the experience is available;
providing an end user an opportunity in the future to receive the experience if the bandwidth is not available; and
providing the end user with the experience if the experience is available.
61. The method defined in Claim 60 further comprising:
starting a browser;
receiving a page from an ASP website; and
downloading a player from the ASP website.
62. The method defined in Claim 60 wherein initiating a request comprises:
multiple end users initiating requests for an enhanced experience; and
a broker determining if resources are available for each request individually.
63. The method defined in Claim 60 wherein the broker is an NSP broker.
64. The method defined in Claim 60 wherein the broker is an ASP broker.
65. A method comprising:
launching a player;
initiating a request for bandwidth necessary for an experience; and
an NSP broker setting aside bandwidth to satisfy the request.
66. A method comprising:
initiating the application;
the application using the library to match the application performance requirements to routines that initiate enhanced performance;

the application initiating the routines;

the application determining whether adequate service is available to provide the enhanced performance; and

causing enhanced performance to be initiated if adequate service is available.

67. The method defined in Claim 66 further comprising presenting options to a user if adequate bandwidth is not available.

68. The method defined in Claim 66 further comprising presenting options to a user if access to an application or network is not available.

69. The method defined in Claim 60 wherein the broker comprises a set of daemons.

70. A method comprising:

a proxy continuously monitoring network traffic generated and received by a client source to determine whether an application attempting to access content that may benefit from an enhanced service;

initiating a service request to an NSP broker if the proxy detects application activity to access content;

the NSP brokering a connection through any intermediary NSP brokers to an ASP broker;

the proxy instructing the NSP broker to provide the service; and
automatically initiating an option for a bandwidth enhanced service.

71. The method defined in Claim 70 further comprising:

the proxy detecting when the application is finished using the service; and

the proxy instructing the NSP broker to discontinue the service.

72. The method defined in Claim 71 further comprising:

the broker removing any associated network reservations; and

the ASP generating a bill for one of the group of end user, employer, experience sponsor, advertiser, or merchant.

73. A method comprising:

a proxy continuously monitoring network traffic generated and received by a client source to determine whether an application attempting to access content that may benefit from an enhanced service;

means for initiating a service request to an NSP broker if the proxy detects application activity to access content;

means for brokering a connection through any intermediary NSP brokers to an ASP broker;

the proxy instructing the NSP broker to provide the service; and

means for automatically initiating an option for a bandwidth enhanced service.

74. A method comprising:

means for launching a player;

means for initiating a request for bandwidth necessary for an experience;

means for determining whether the experience is available;

means for providing an end user an opportunity in the future to receive the experience if the bandwidth is not available; and

means for providing the end user with the experience if the experience is available.

75. A method comprising launching a player;

means for initiating a request for bandwidth necessary for an experience; and

means for setting aside bandwidth to satisfy the request.

76. A method comprising:

means for initiating the application;

means for using the library to match the application performance requirements to routines that initiate enhanced performance;

means for initiating the routines;

means for determining whether adequate service is available to provide the enhanced performance; and

means for causing enhanced performance to be initiated if adequate service is available.

77. An article of manufacture having one or more recordable medium with executable instructions stored thereon which, when executed by a processing device, cause the processing device to:

- launch a player;
- initiate a request for bandwidth necessary for an experience;
- determine whether the experience is available;
- provide an end user an opportunity in the future to receive the experience if the bandwidth is not available; and
- provide the end user with the experience if the experience is available.

78. An article of manufacture having one or more recordable medium with executable instructions stored thereon which, when executed by a processing device, cause the processing device to:

- initiate a request for bandwidth necessary for an experience; and
- set aside bandwidth to satisfy the request.

79. An article of manufacture having one or more recordable medium with executable instructions stored thereon which, when executed by a processing device, cause the processing device to:

- initiate the application;
- use the library to match the application performance requirements to routines that initiate enhanced performance;
- initiate the routines;
- determine whether adequate service is available to provide the enhanced performance;
- and
- cause enhanced performance to be initiated if adequate service is available.

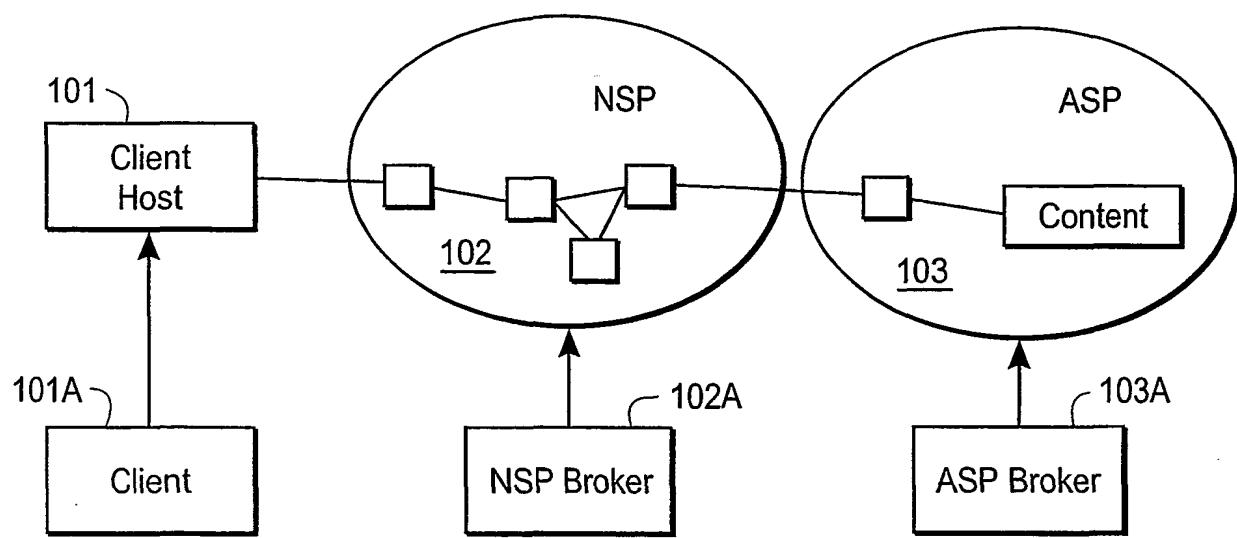


FIG. 1A

2/35

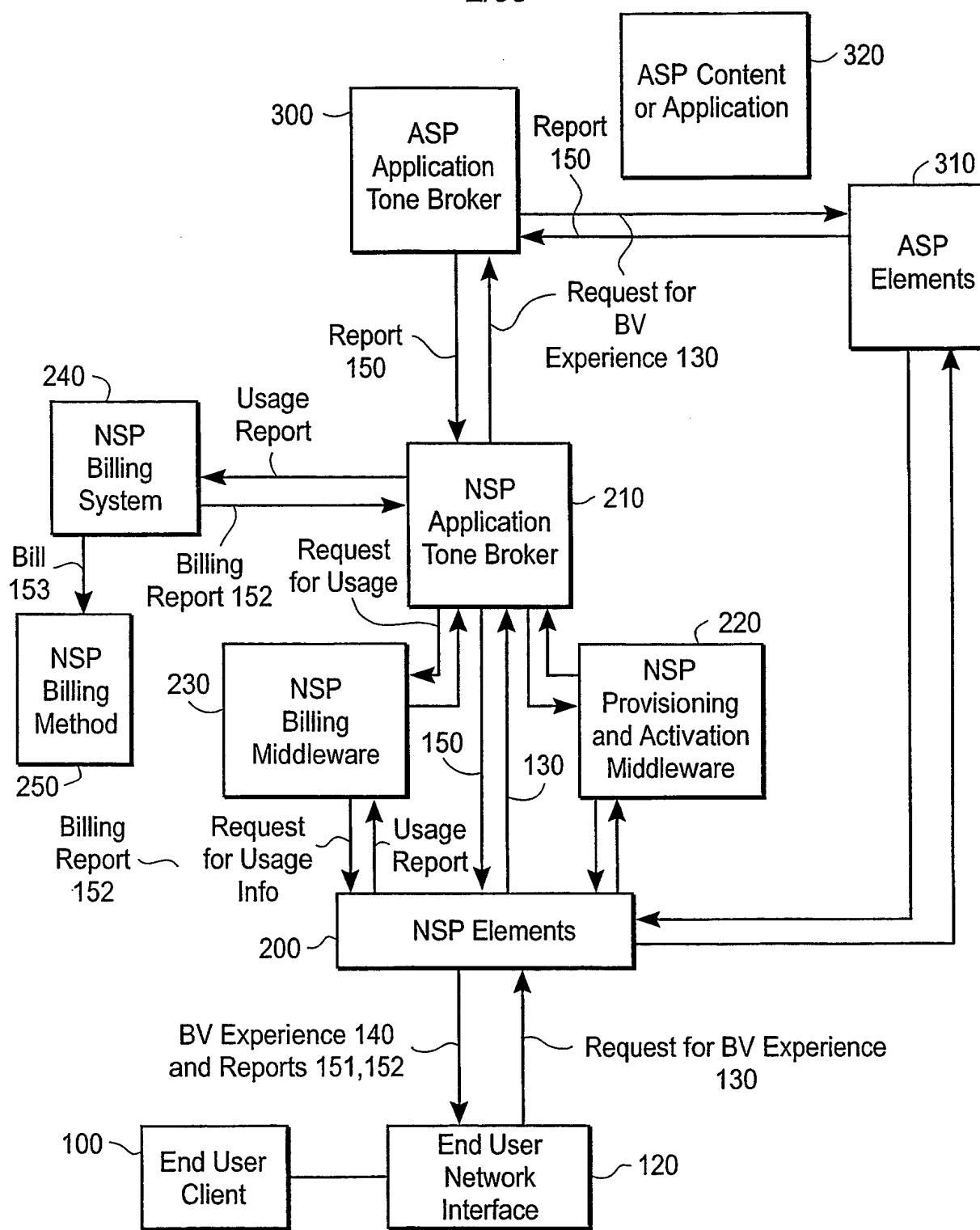


FIG. 1B

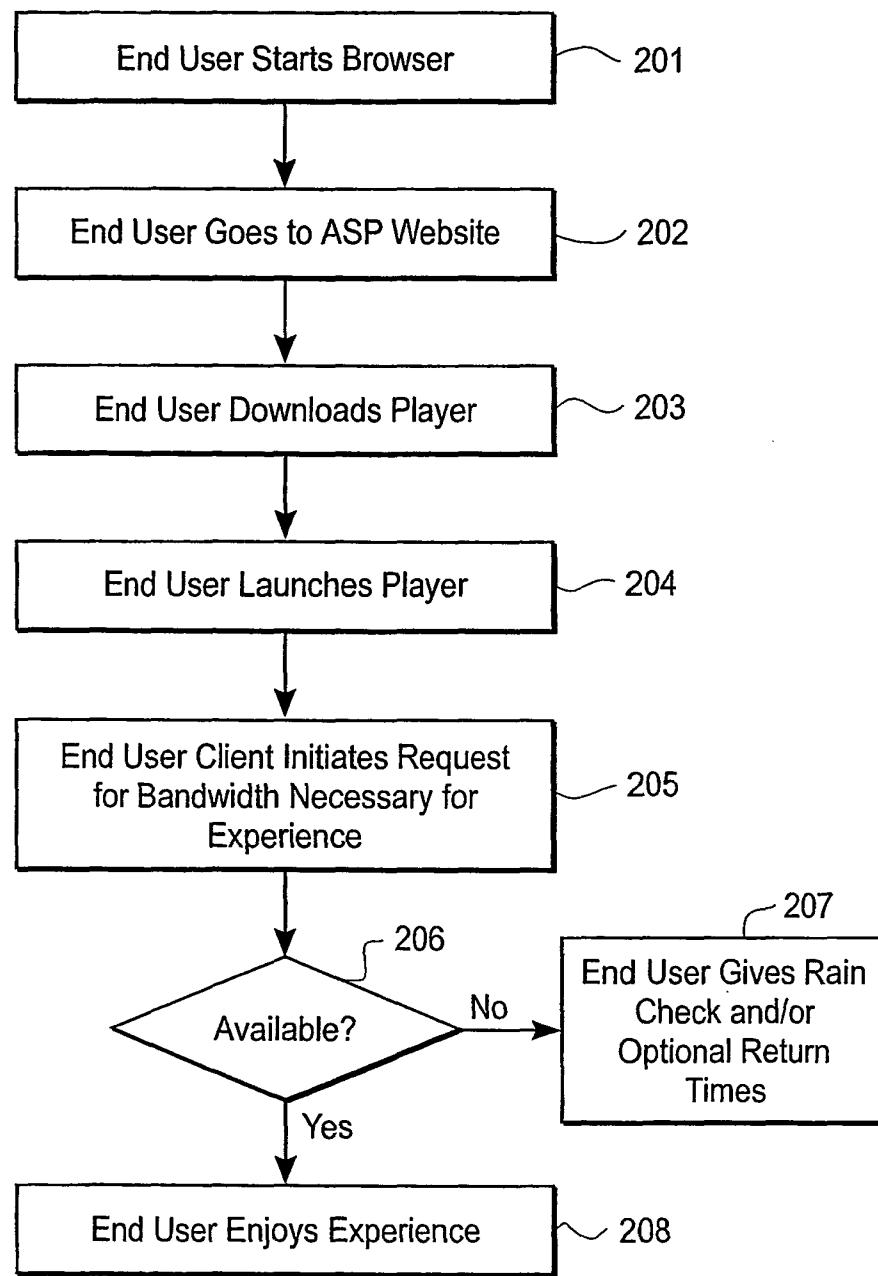


FIG. 2A

4/35

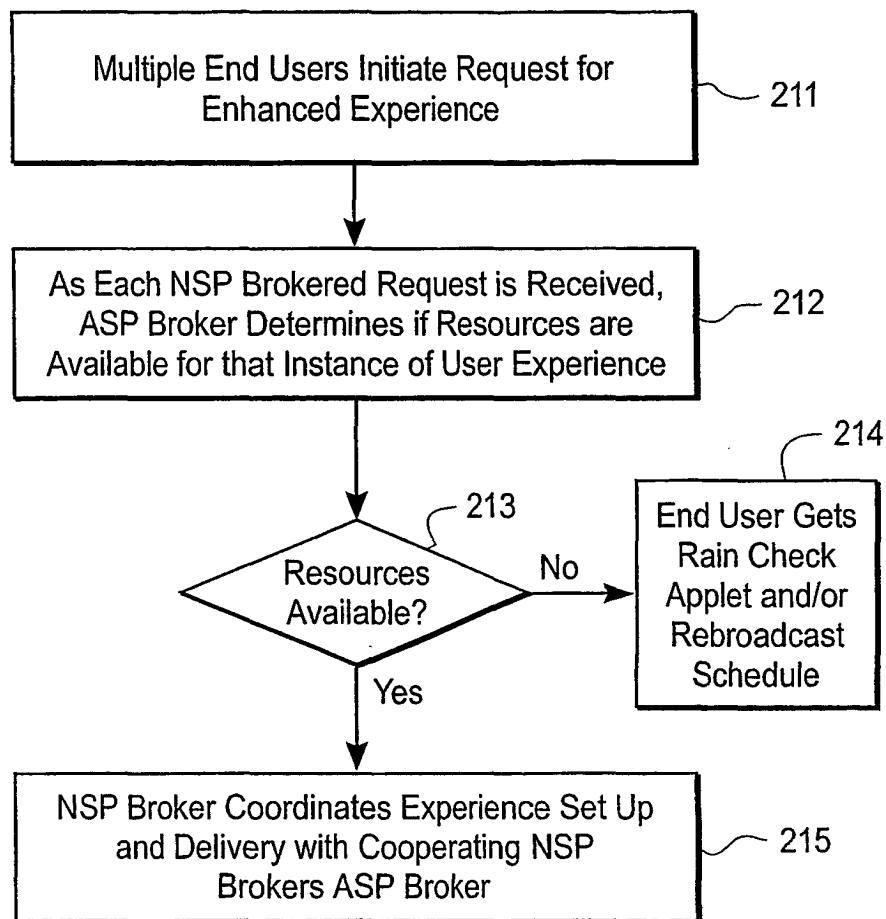


FIG. 2B

5/35

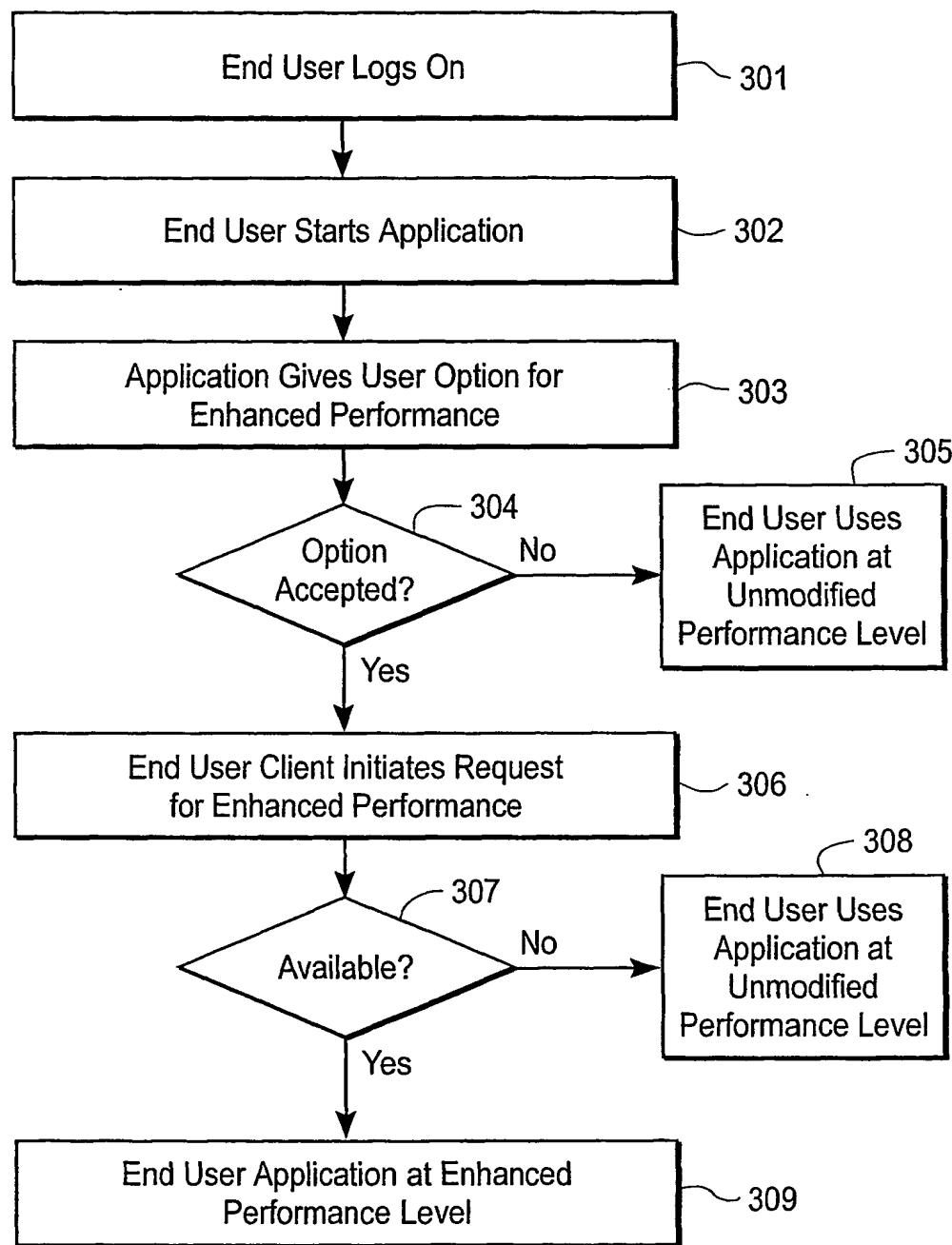


FIG. 3

6/35

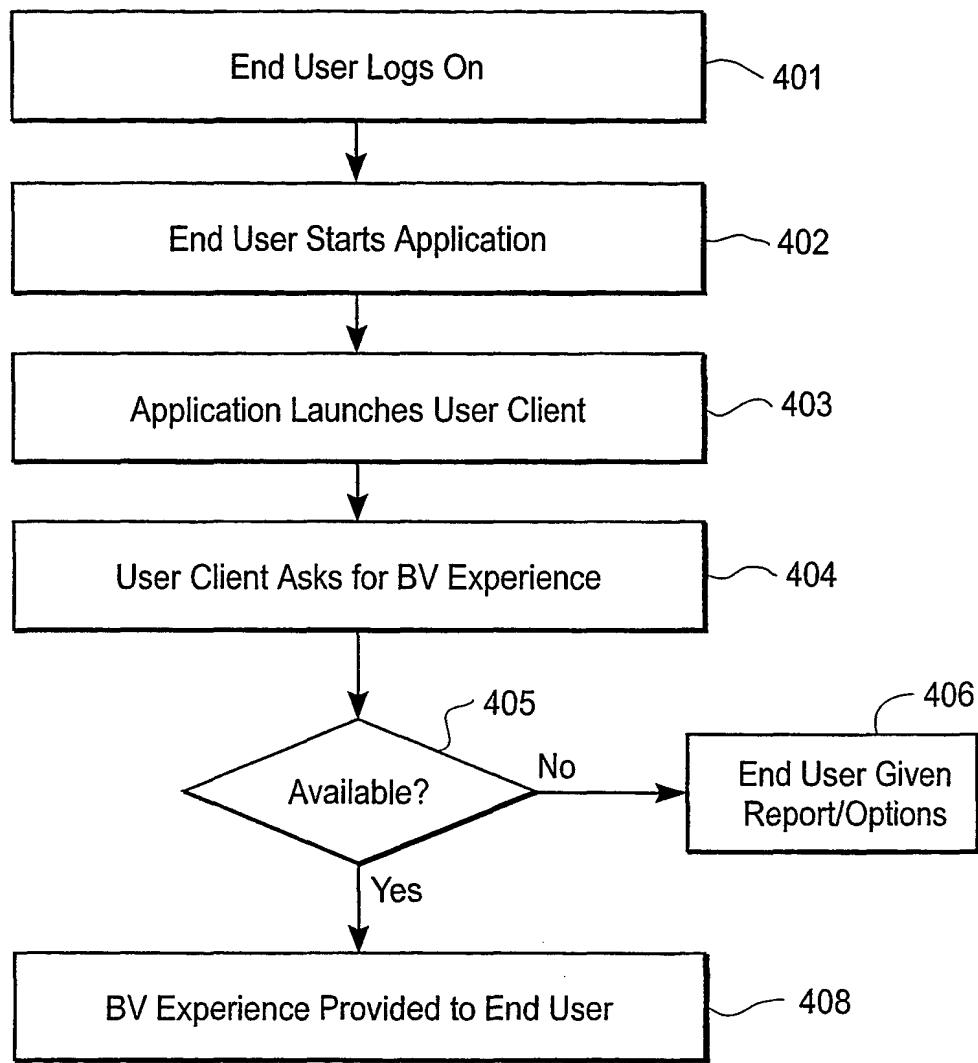


FIG. 4

7/35

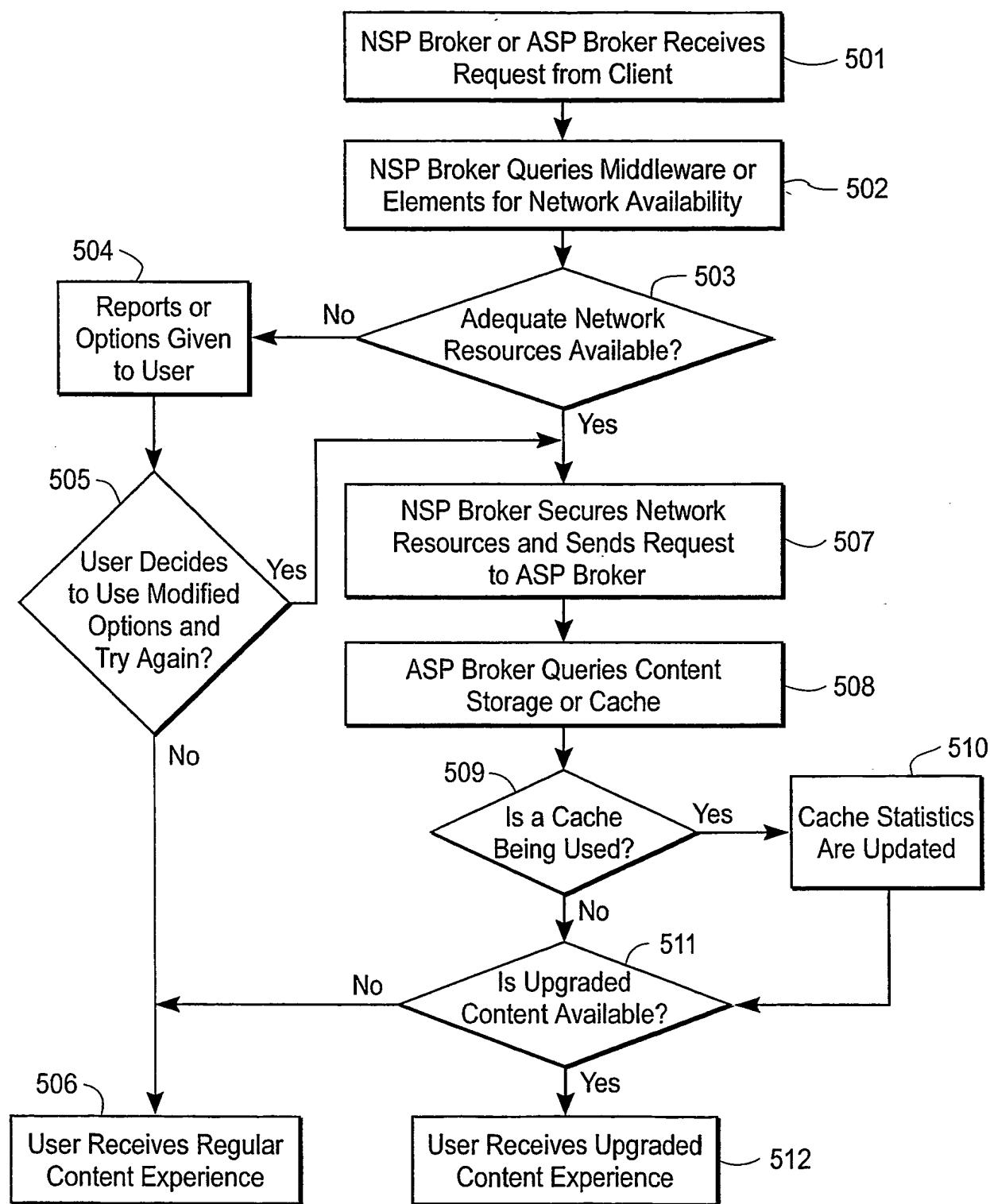


FIG. 5

8/35

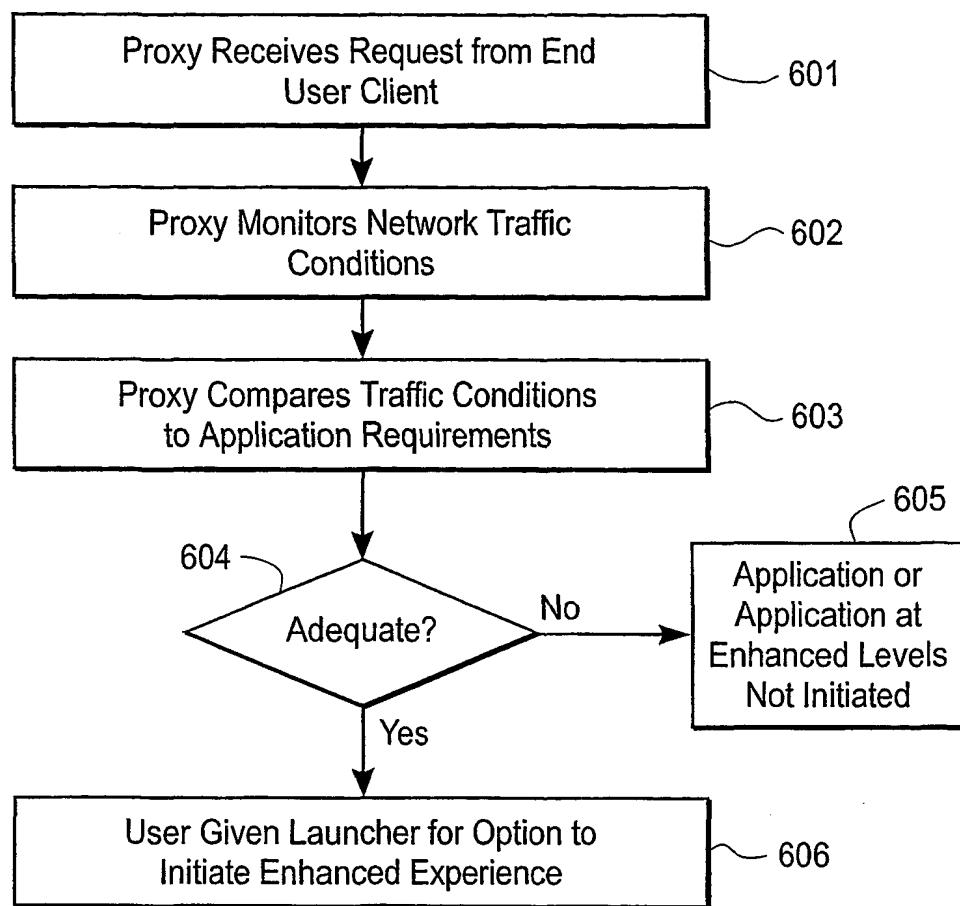


FIG. 6

9/35

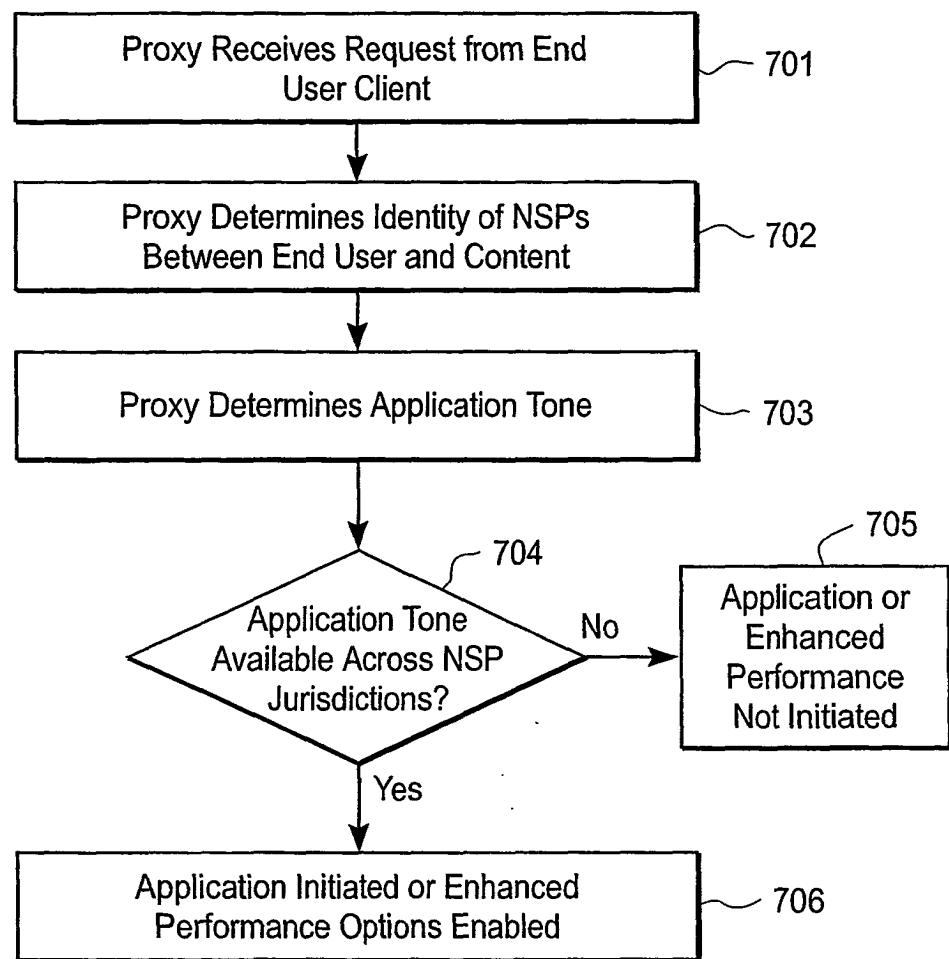


FIG. 7

10/35

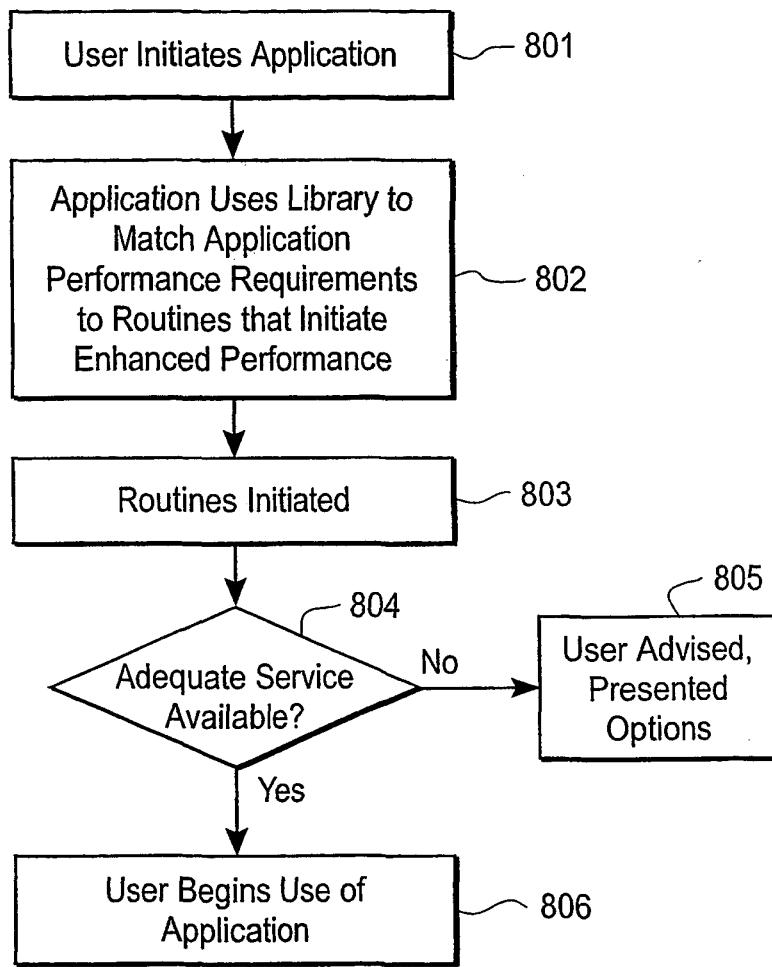


FIG. 8

11/35

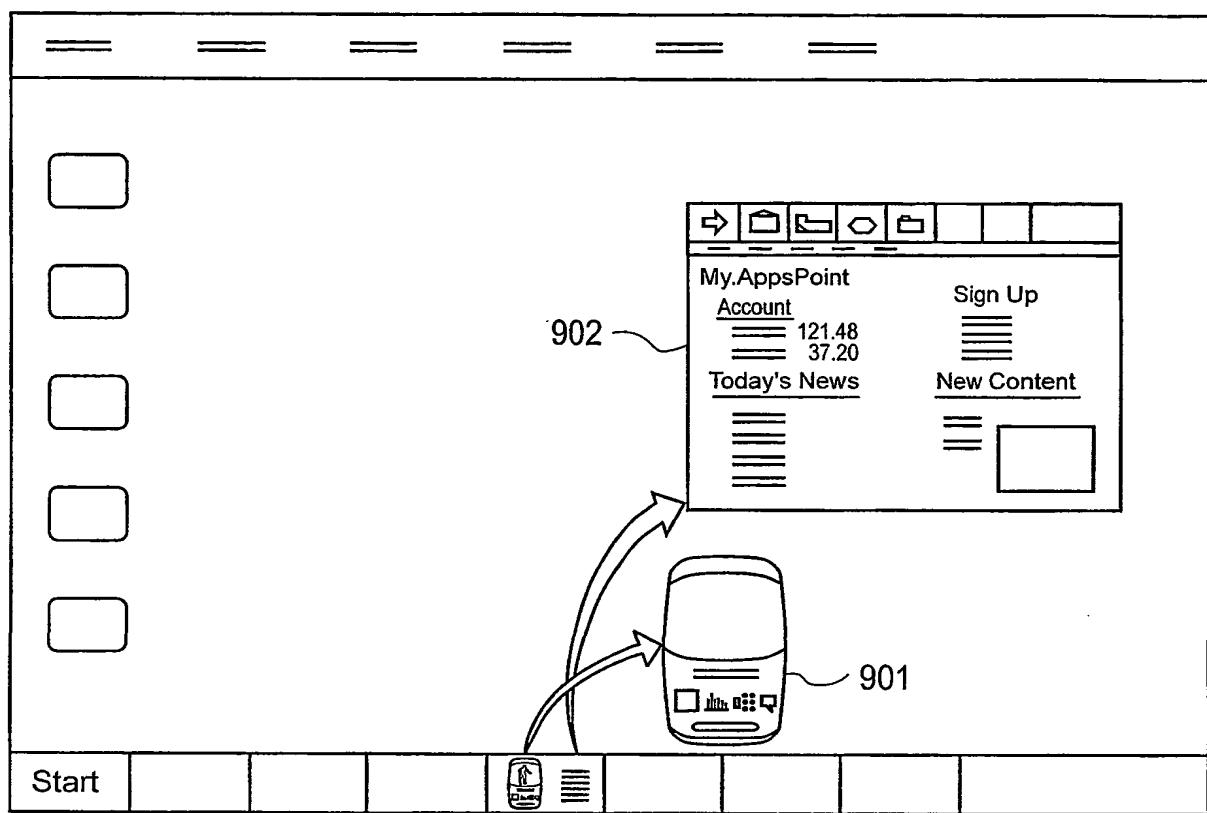


FIG. 9

12/35

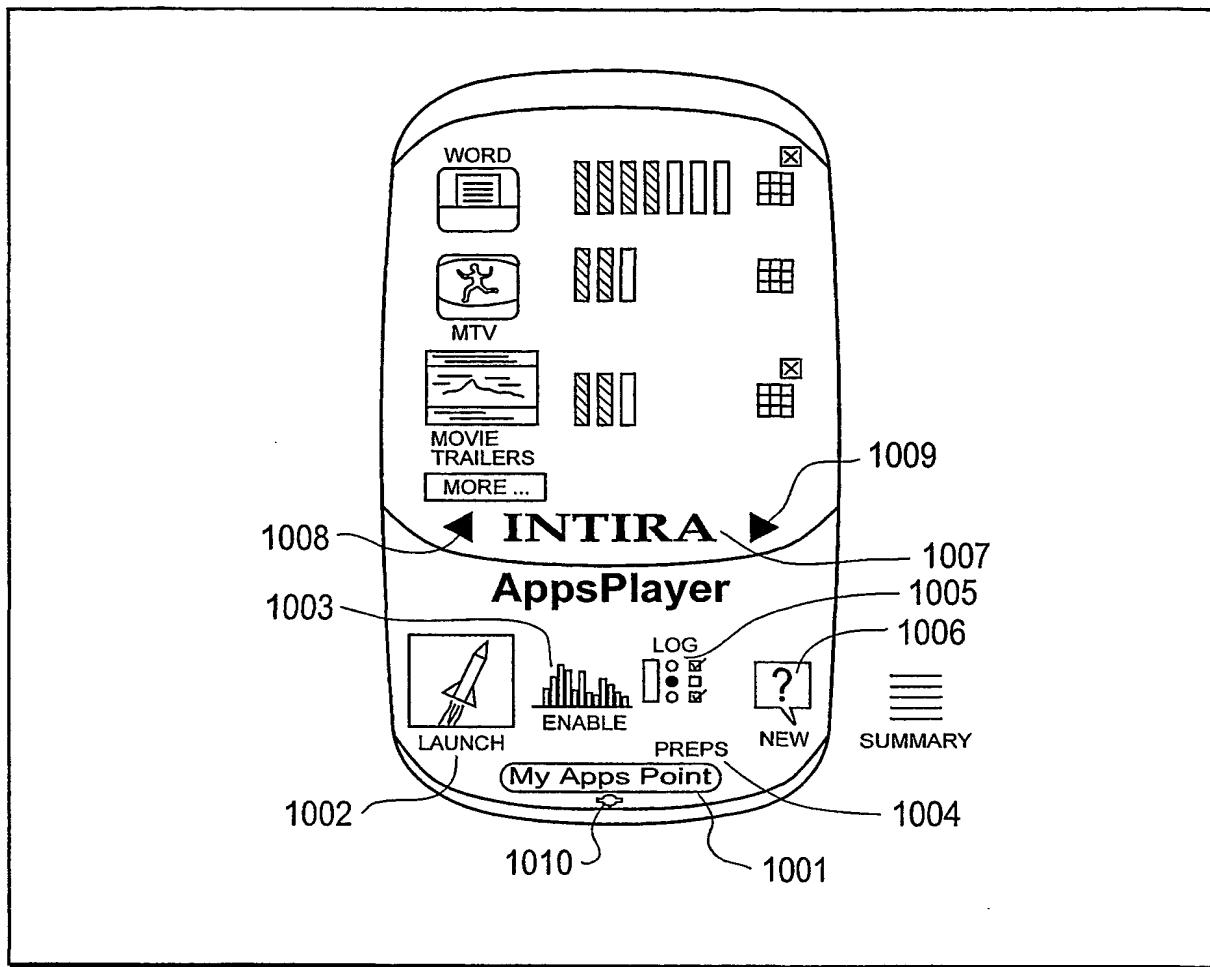
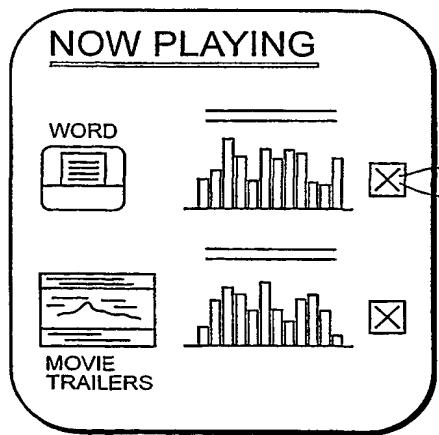


FIG. 10

Other Panes:



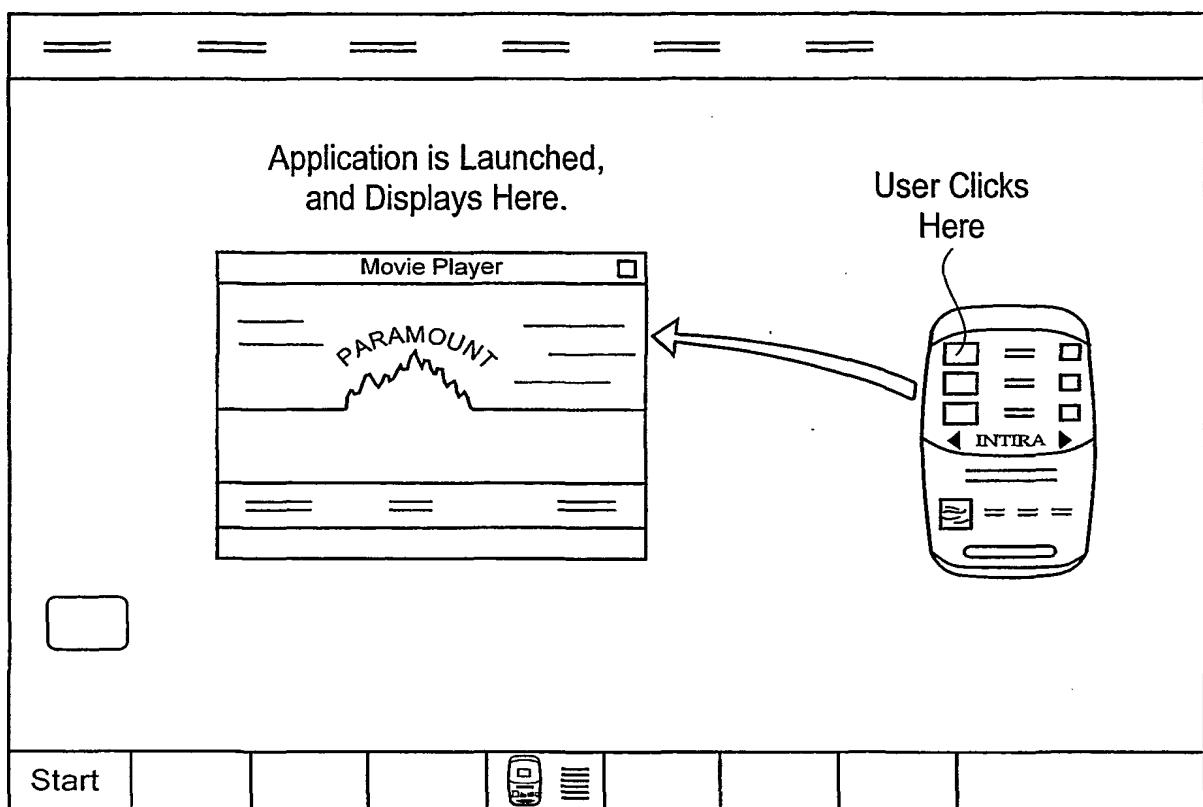
1201

This is the "enable" button. When checked, the already running application receives improved service.

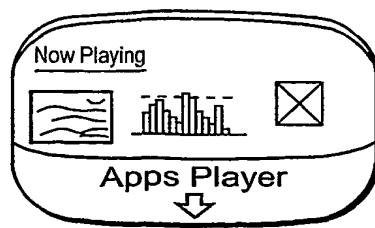
In most cases, only one application at a time may receive AppsPoint service.

FIG. 12

13/35

"Launching an Application"**FIG. 13**Summary Pane

DATE/TIME	CONTENT	SP	\$\$
====	====	====	====
====	====	====	====
====	====	====	====
====	====	====	====
Running	Movie	INTIRA	3.50

Shrunken**FIG. 11****FIG. 14**

14/35

"MY.APPSPPOINT.COM"

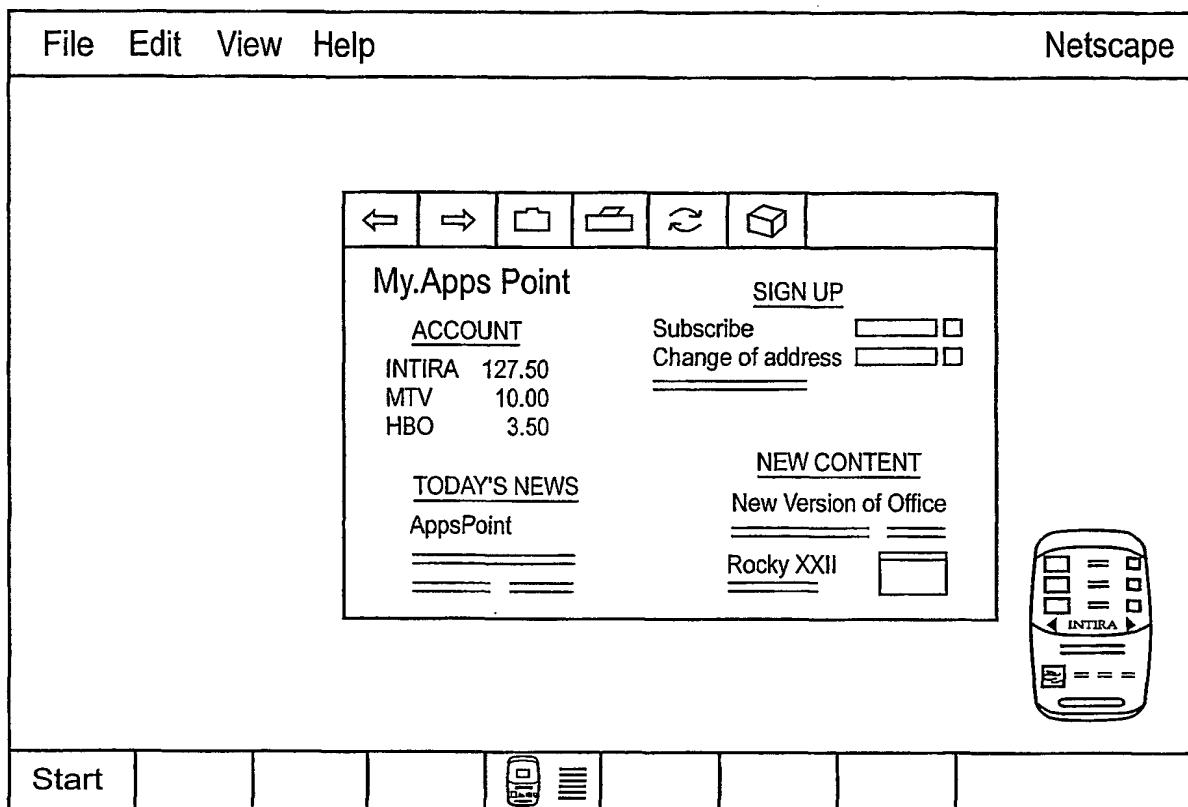


FIG. 15

15/35

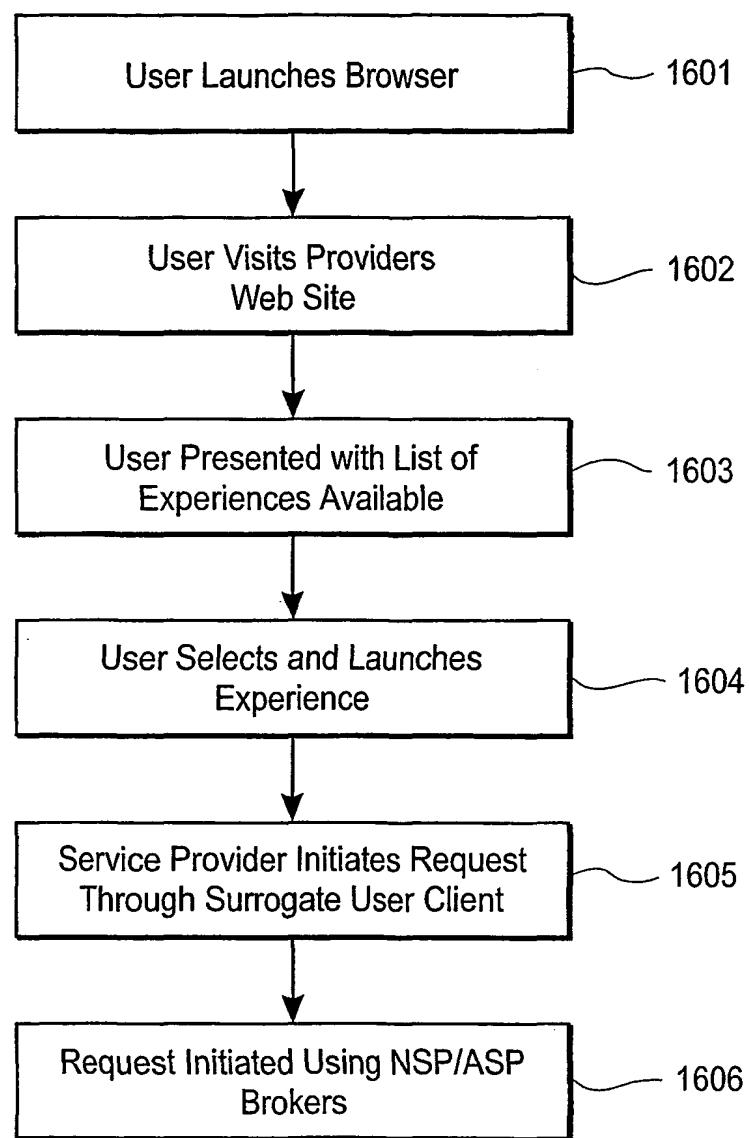


FIG. 16

16/35

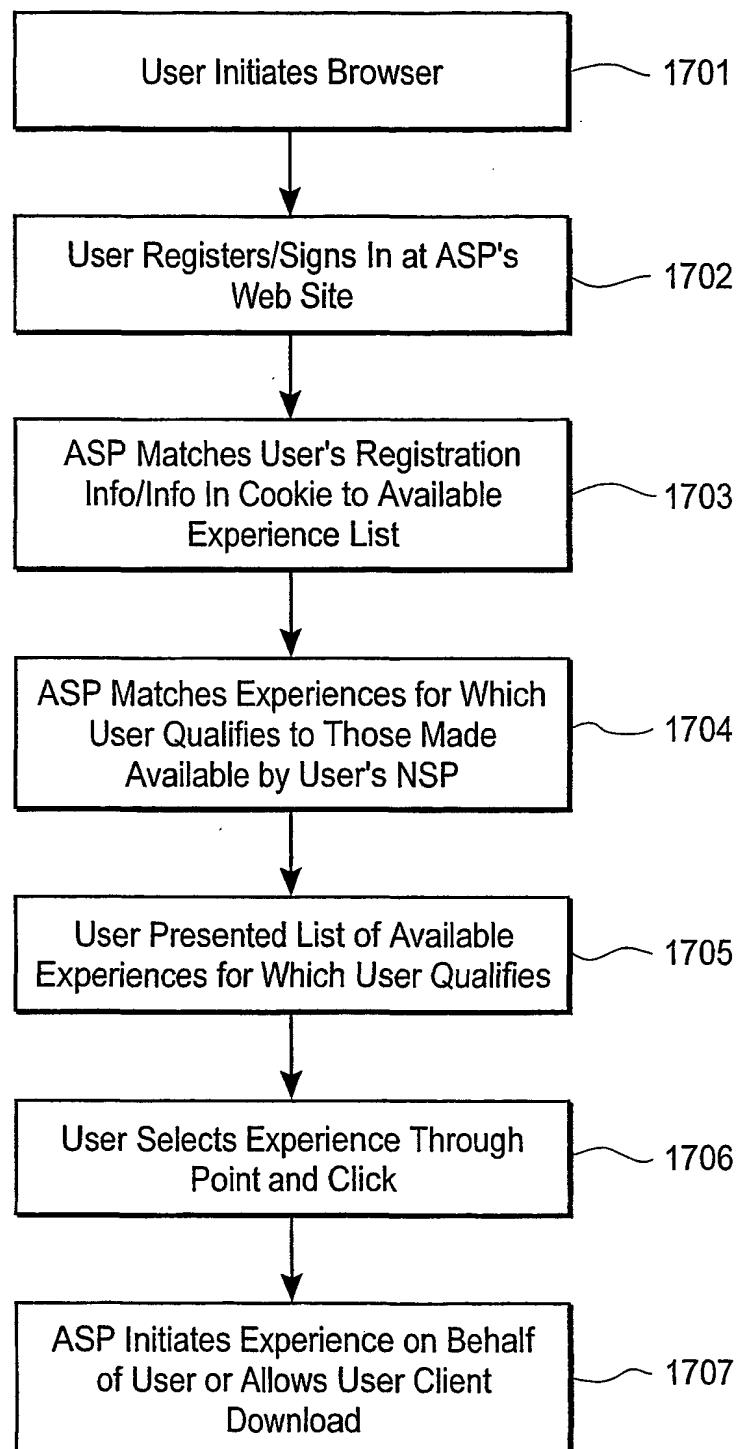


FIG. 17

17/35

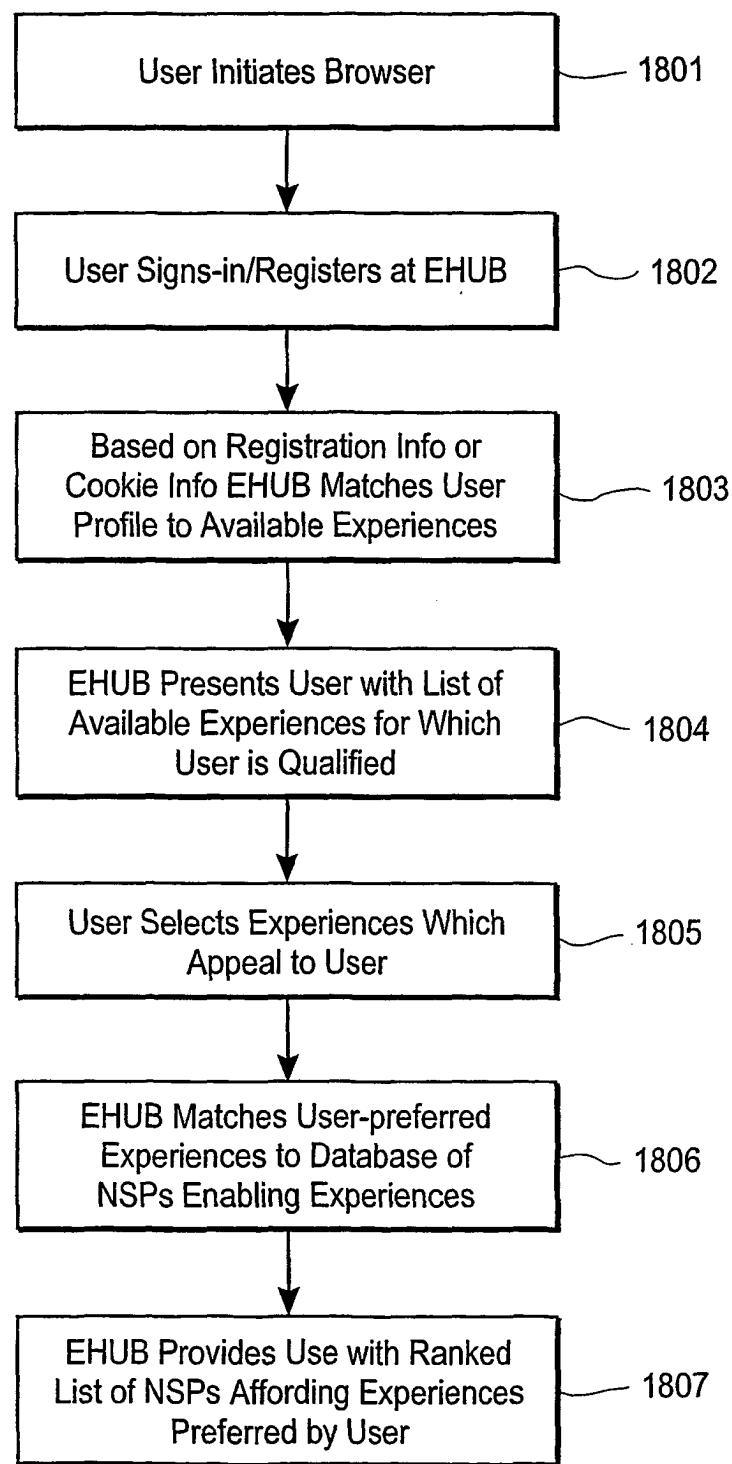


FIG. 18

18/35

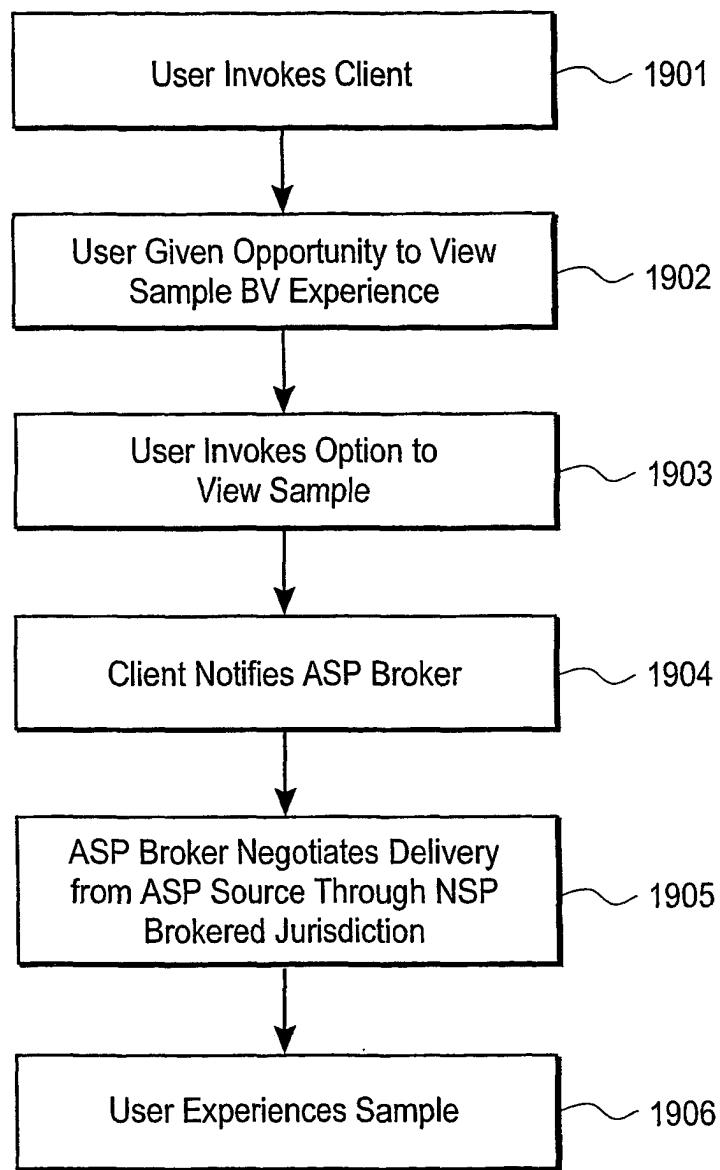


FIG. 19

19/35

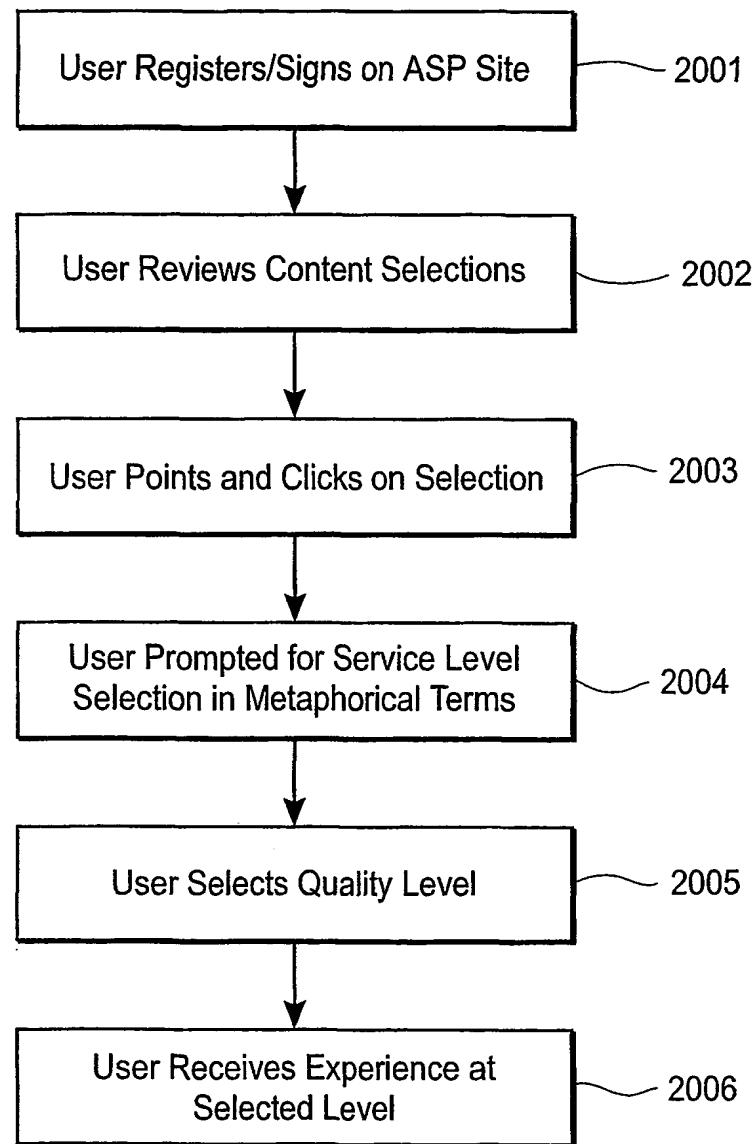


FIG. 20

20/35

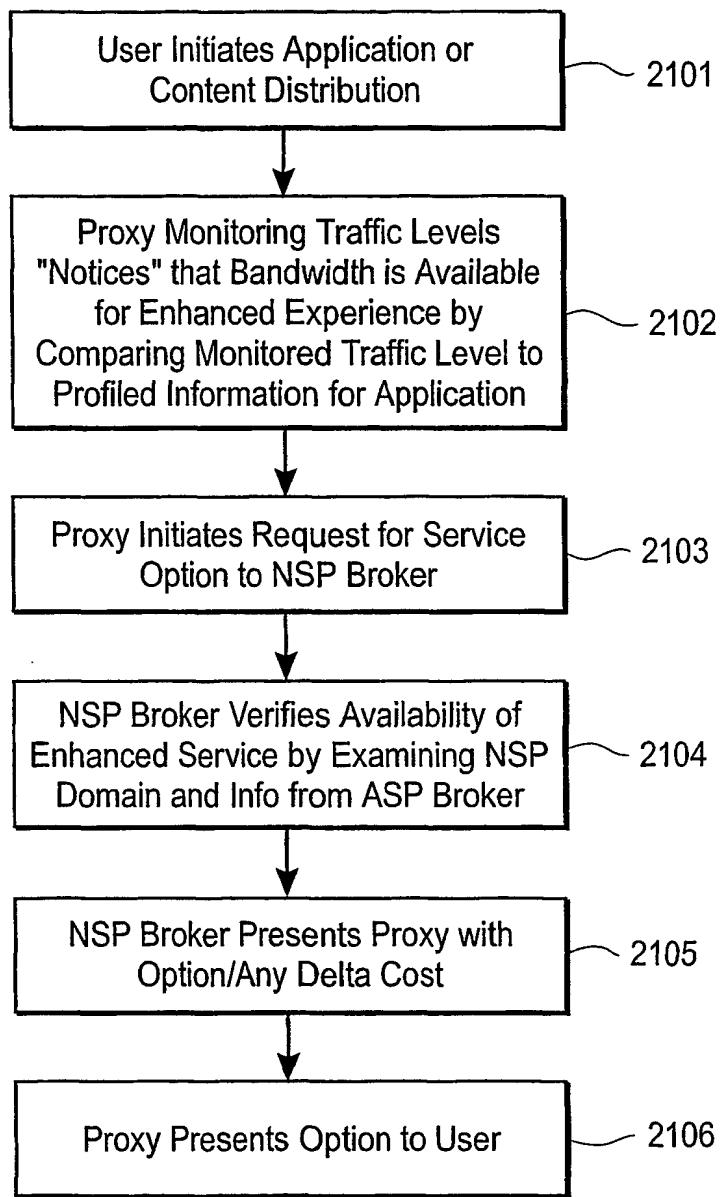


FIG. 21

21/35

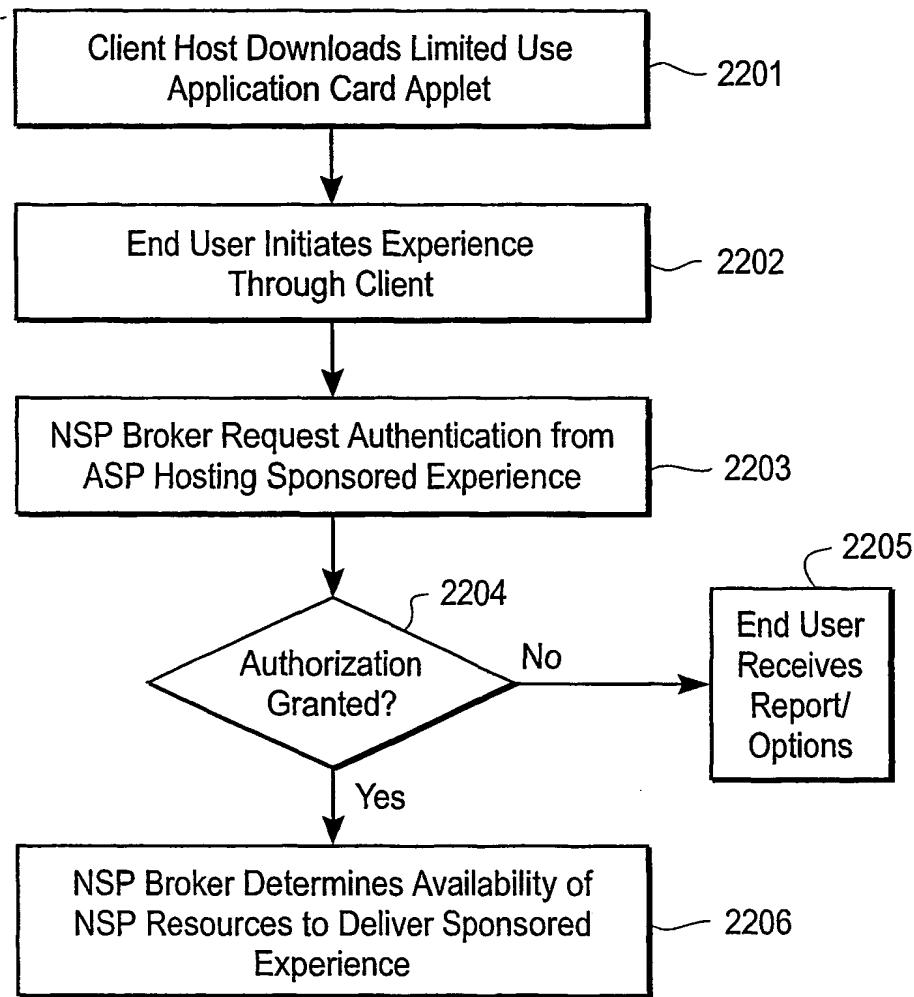


FIG. 22

22/35

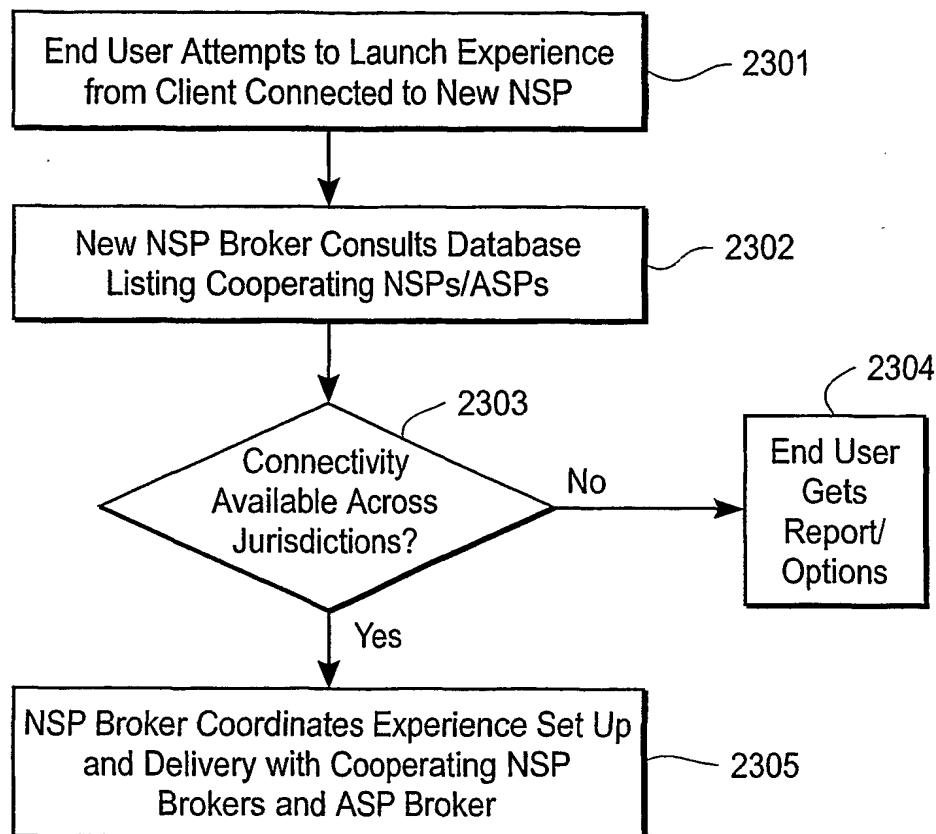


FIG. 23

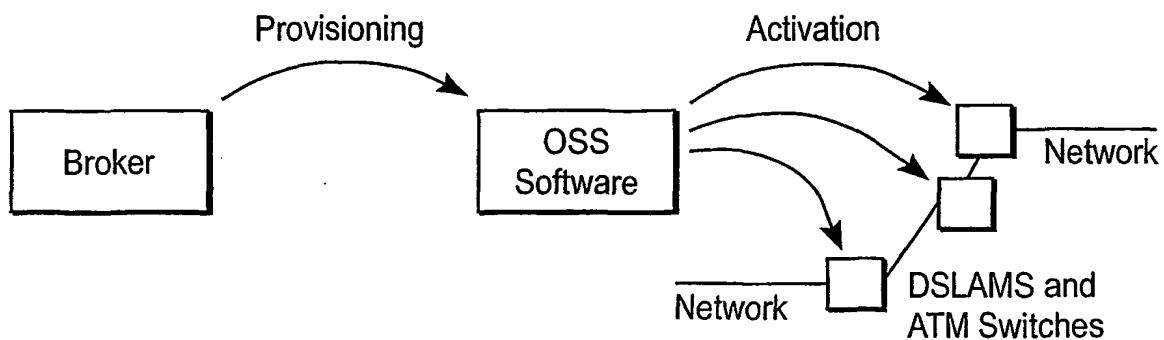


FIG. 24

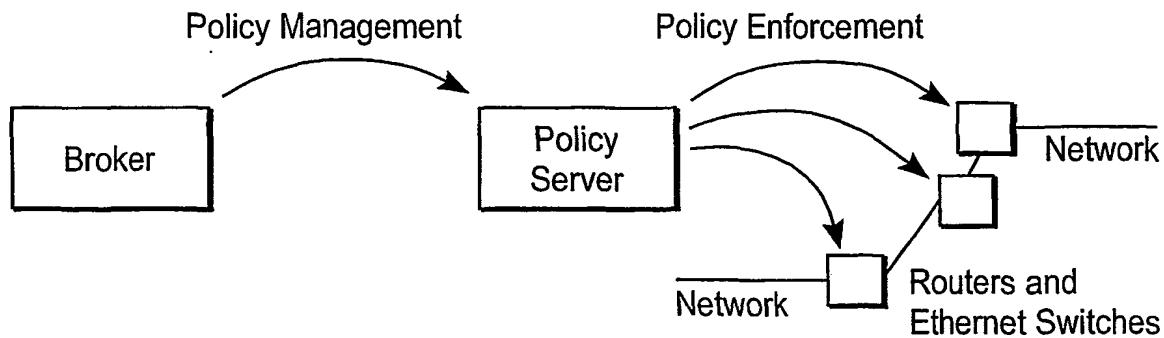


FIG. 29

24/35

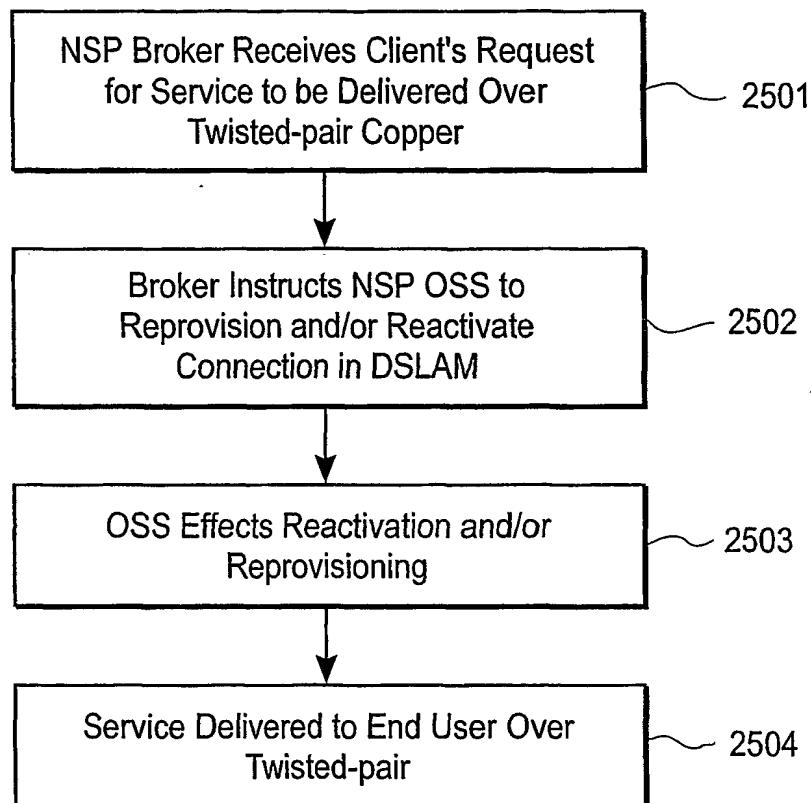


FIG. 25

25/35

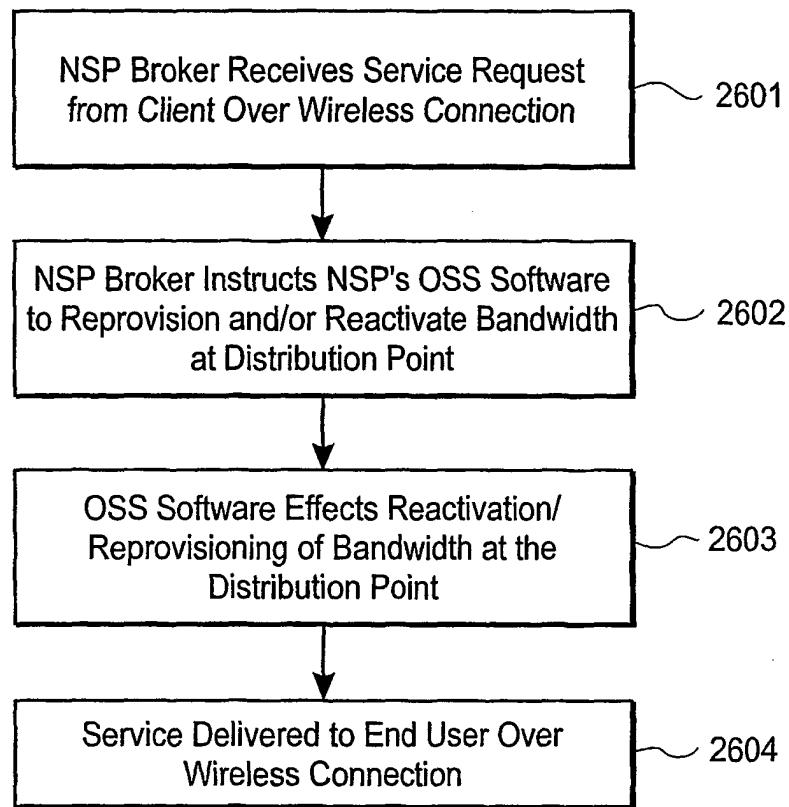


FIG. 26

26/35

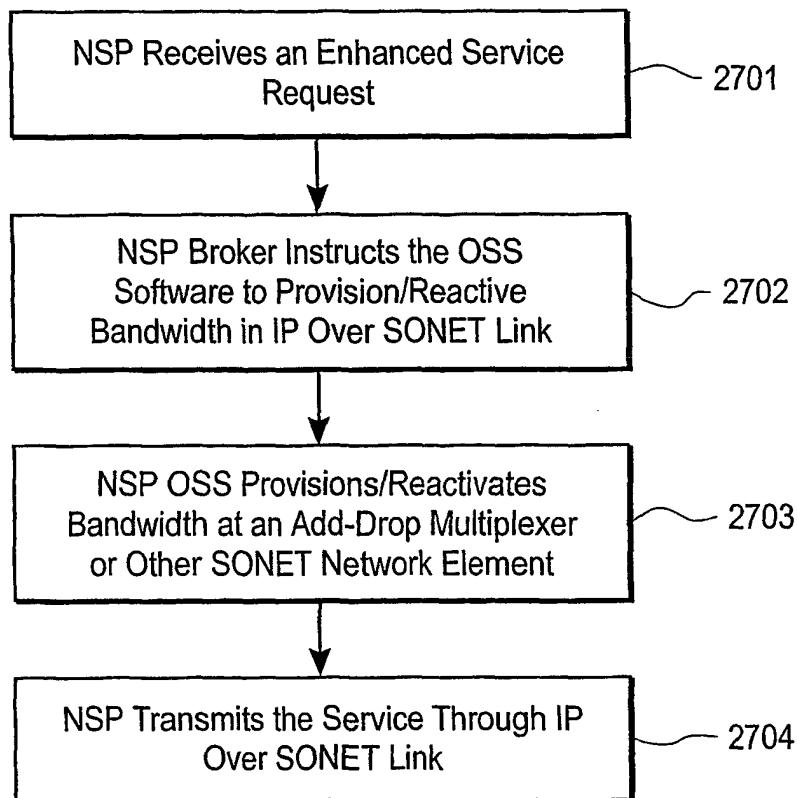


FIG. 27

27/35

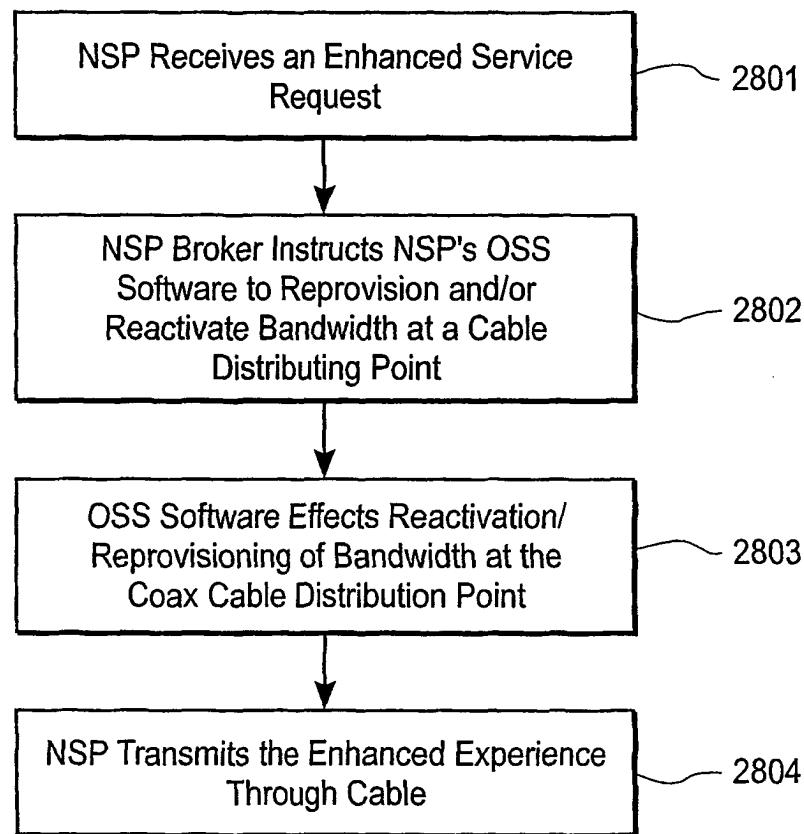


FIG. 28

28/35

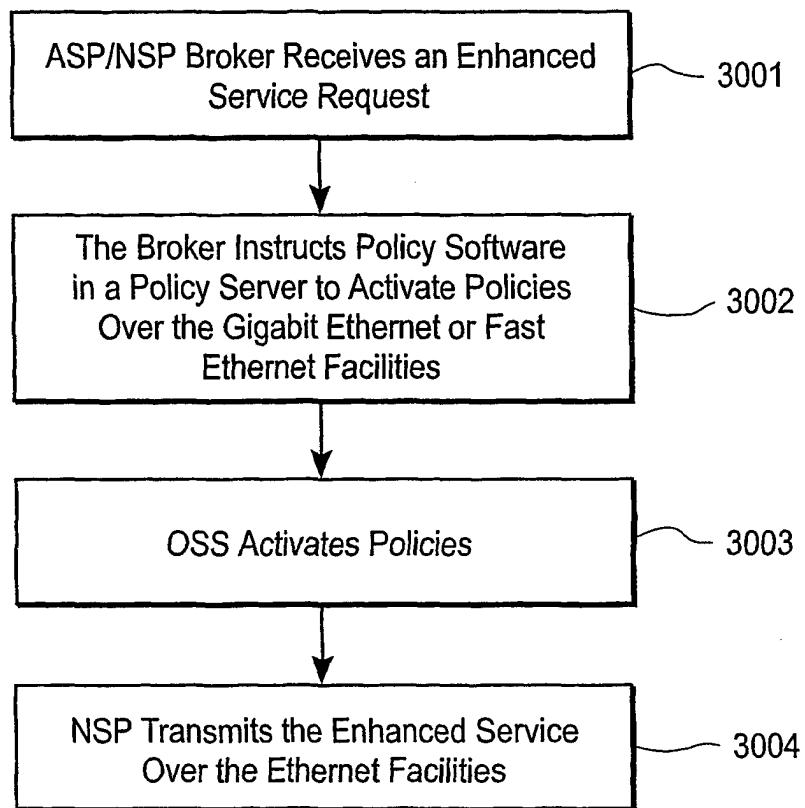


FIG. 30

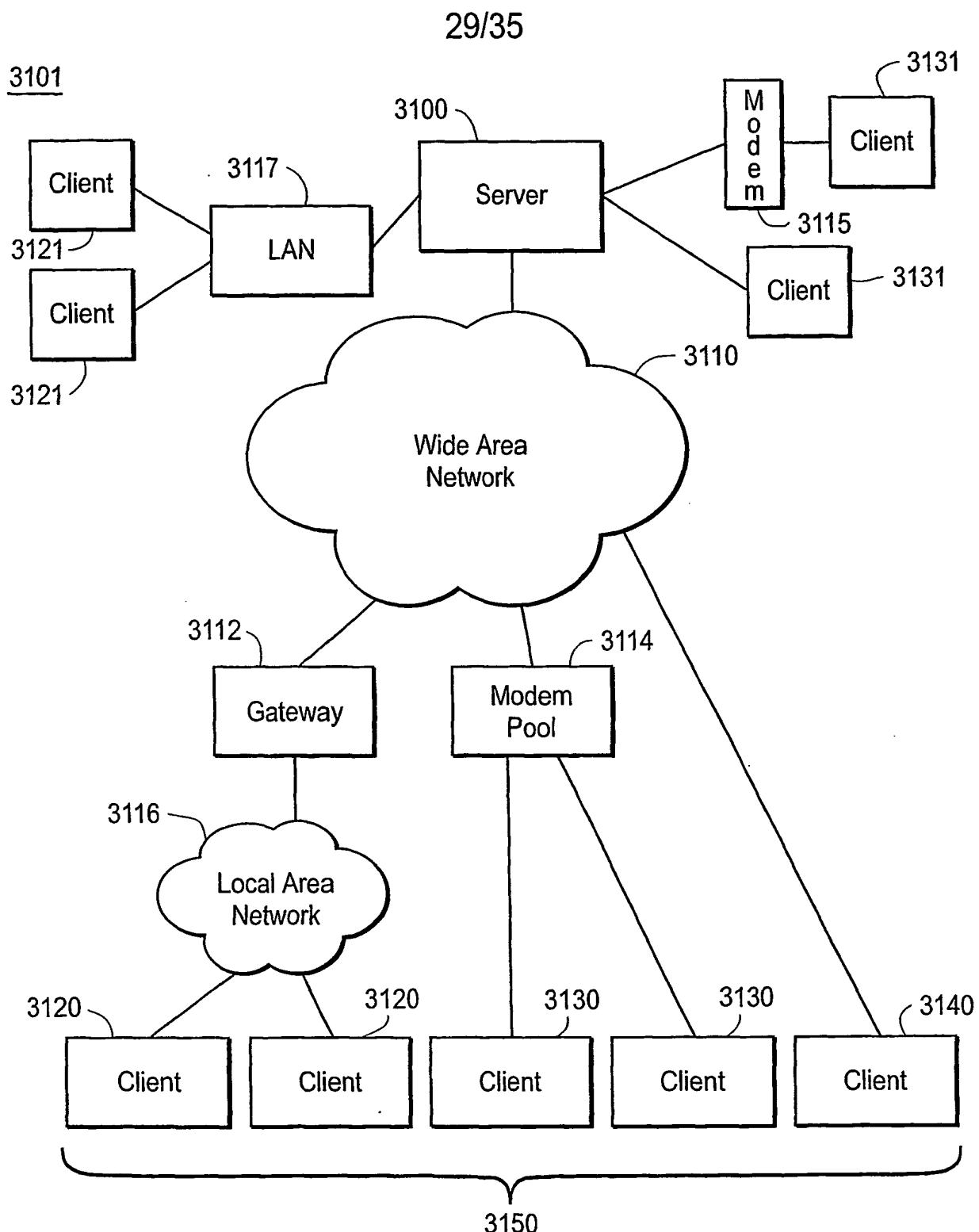


FIG. 31A

30/35

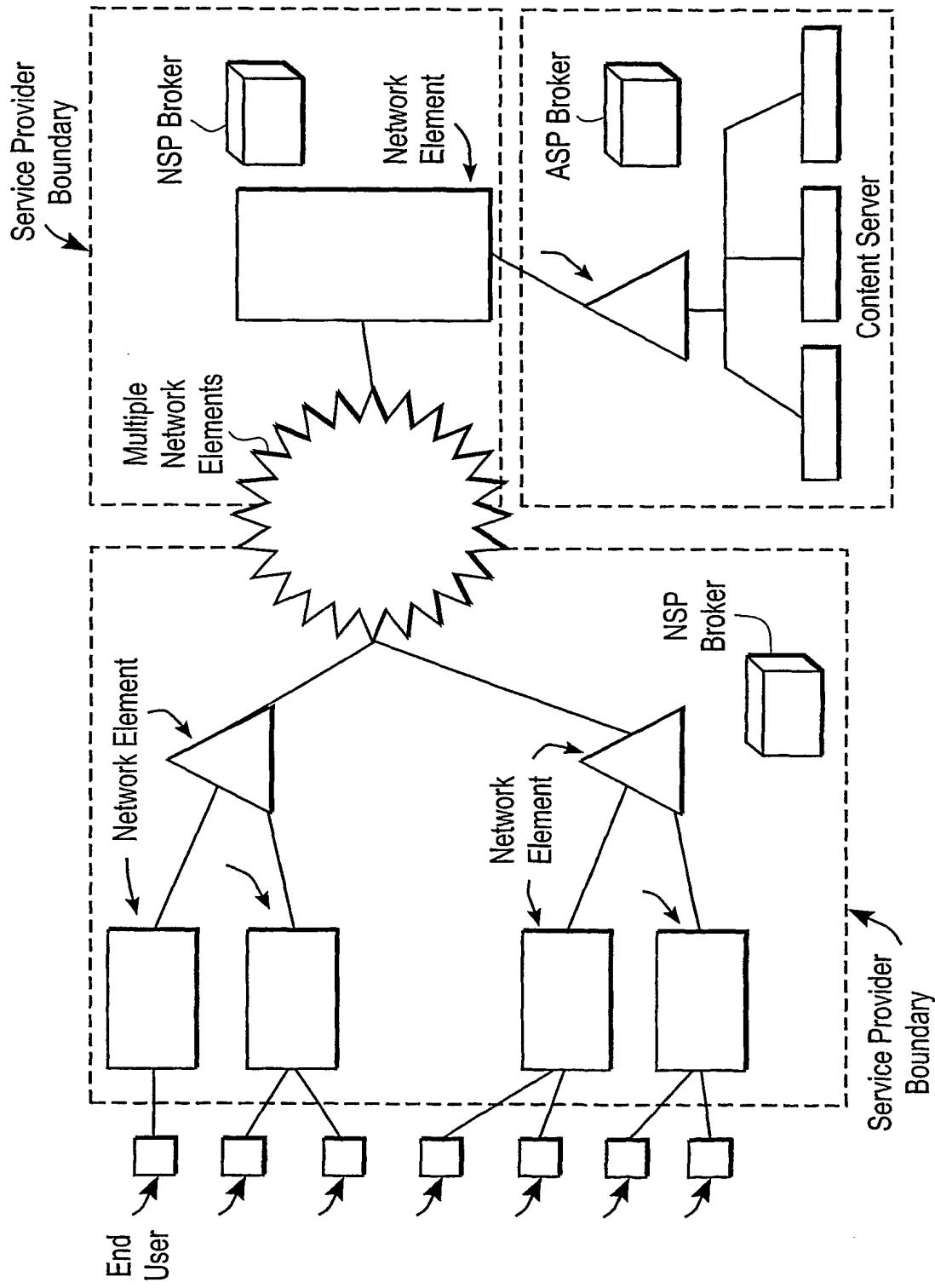


FIG. 31B

31/35

MULTIPLE PHYSICAL PATHS
(Carriers Are Not Necessarily Enabled)

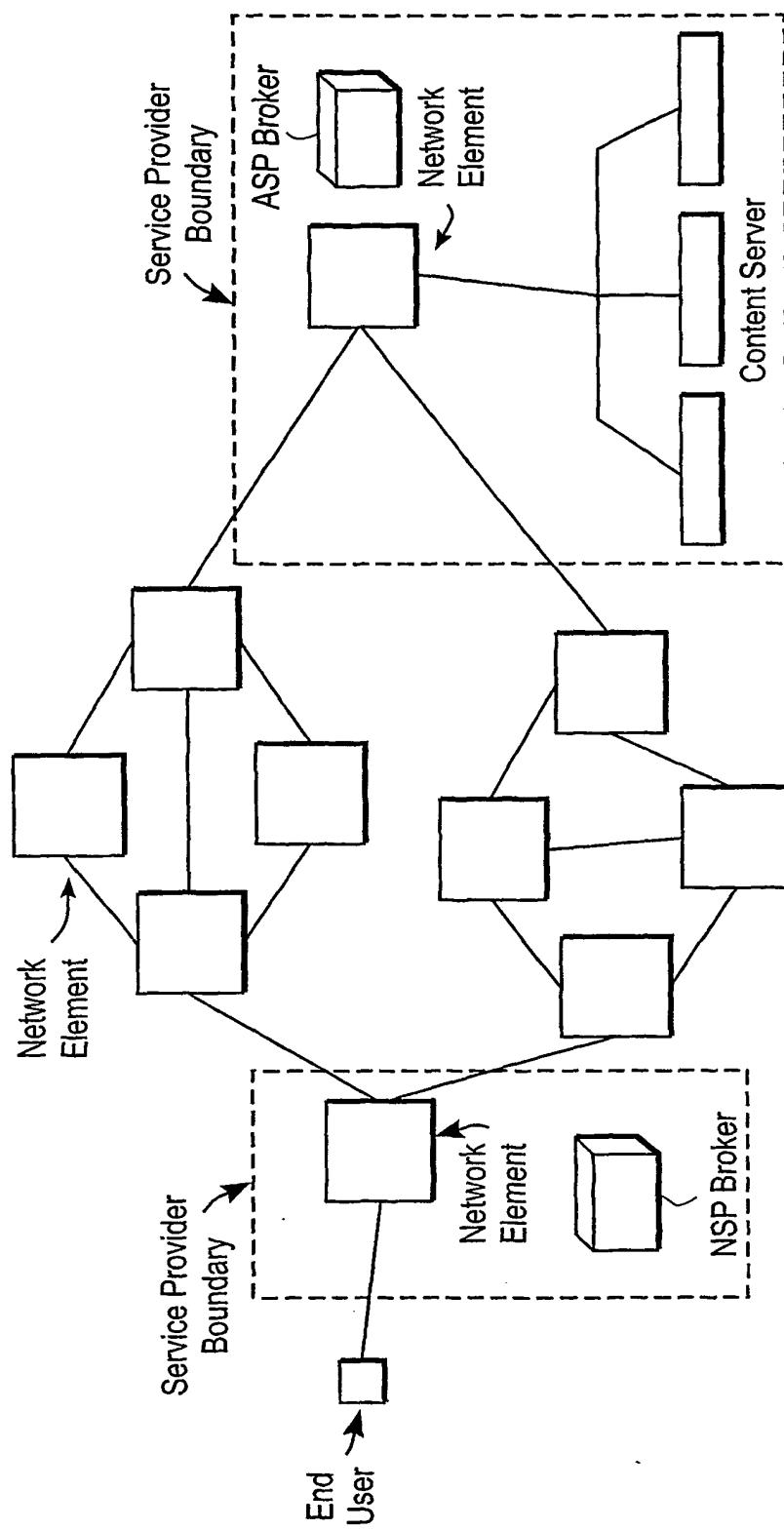


FIG. 31C

32/35

MULTIPLE CLIENT COLLABORATION

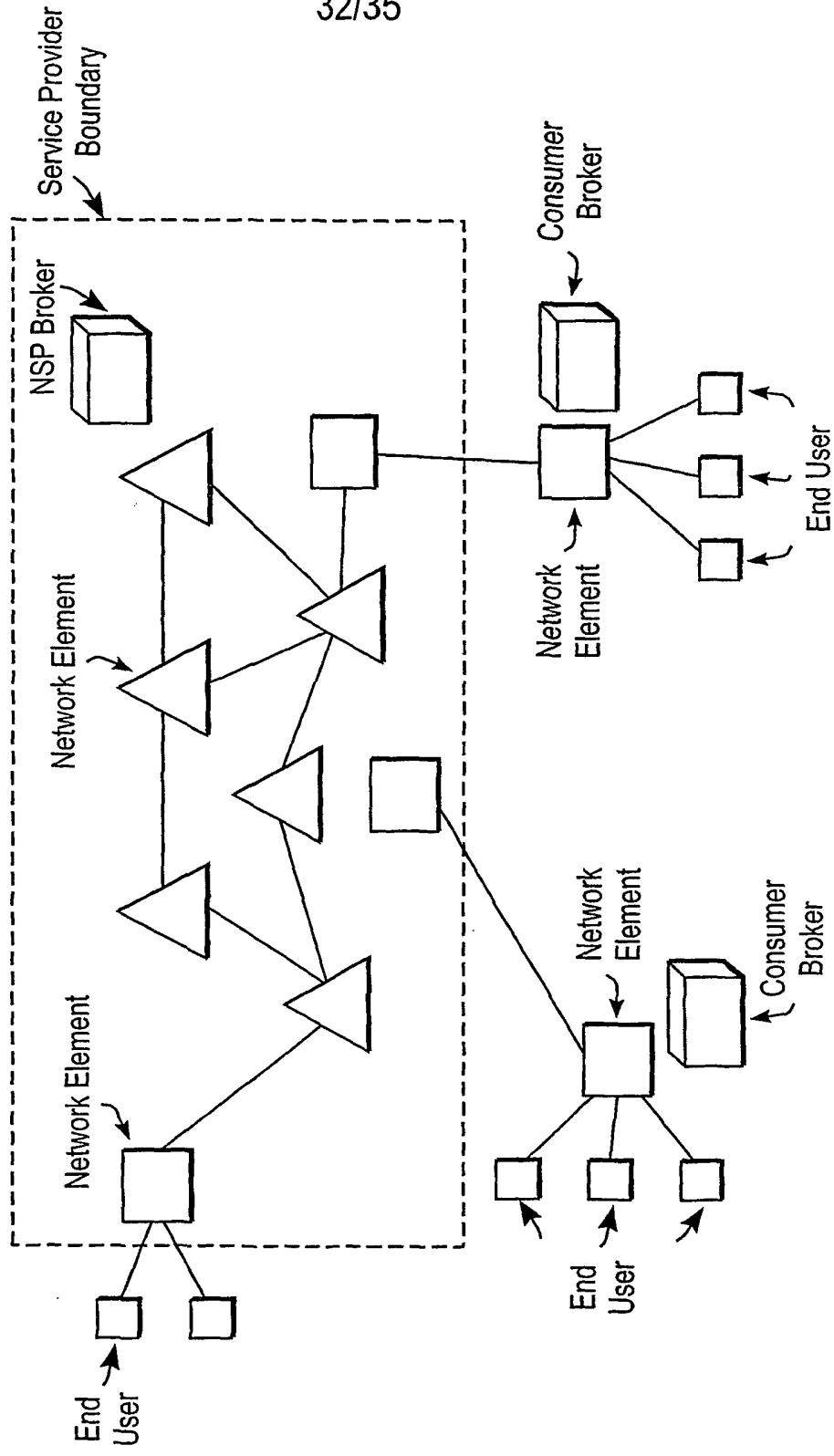


FIG. 31D

33/35

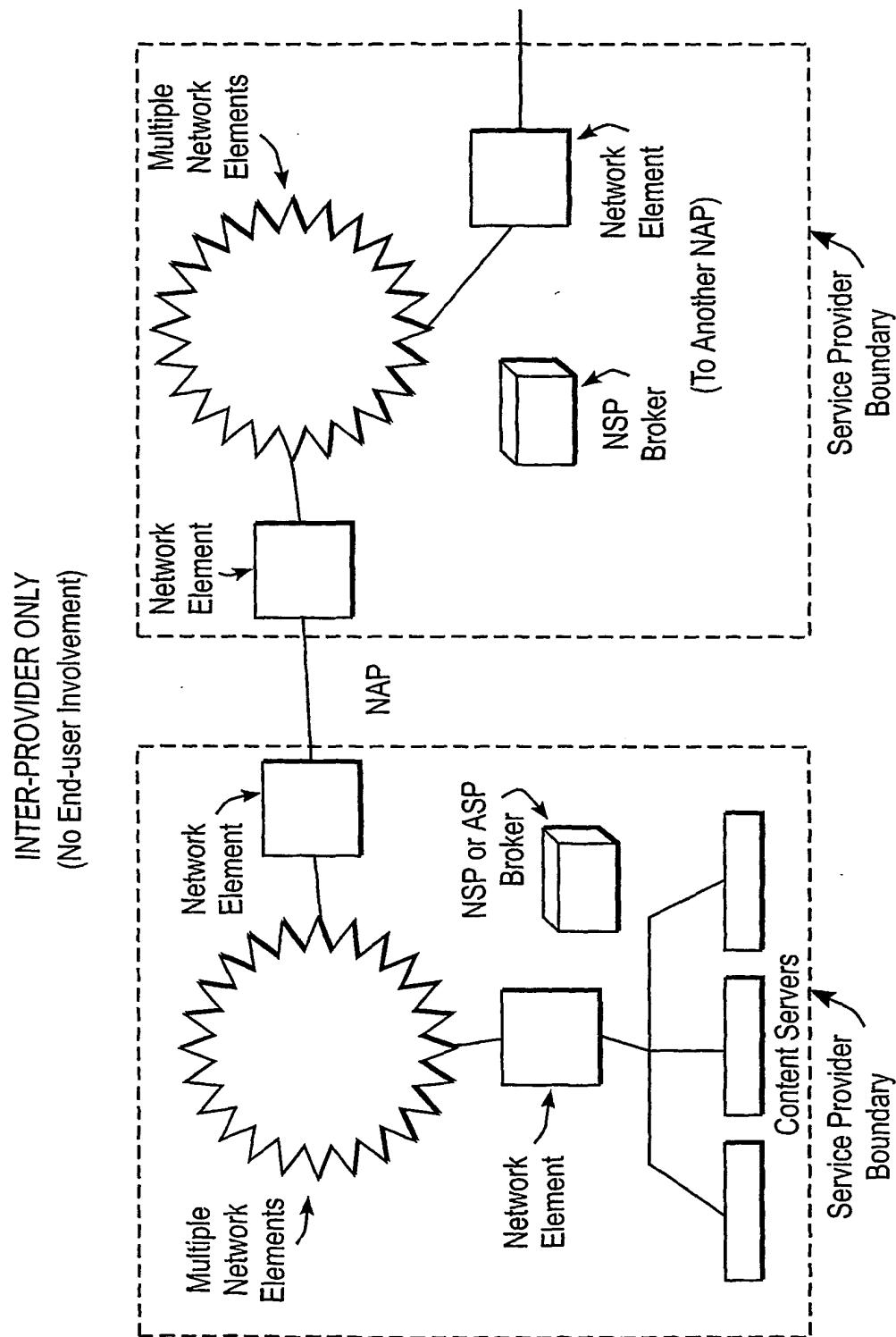


FIG. 31E

34/35

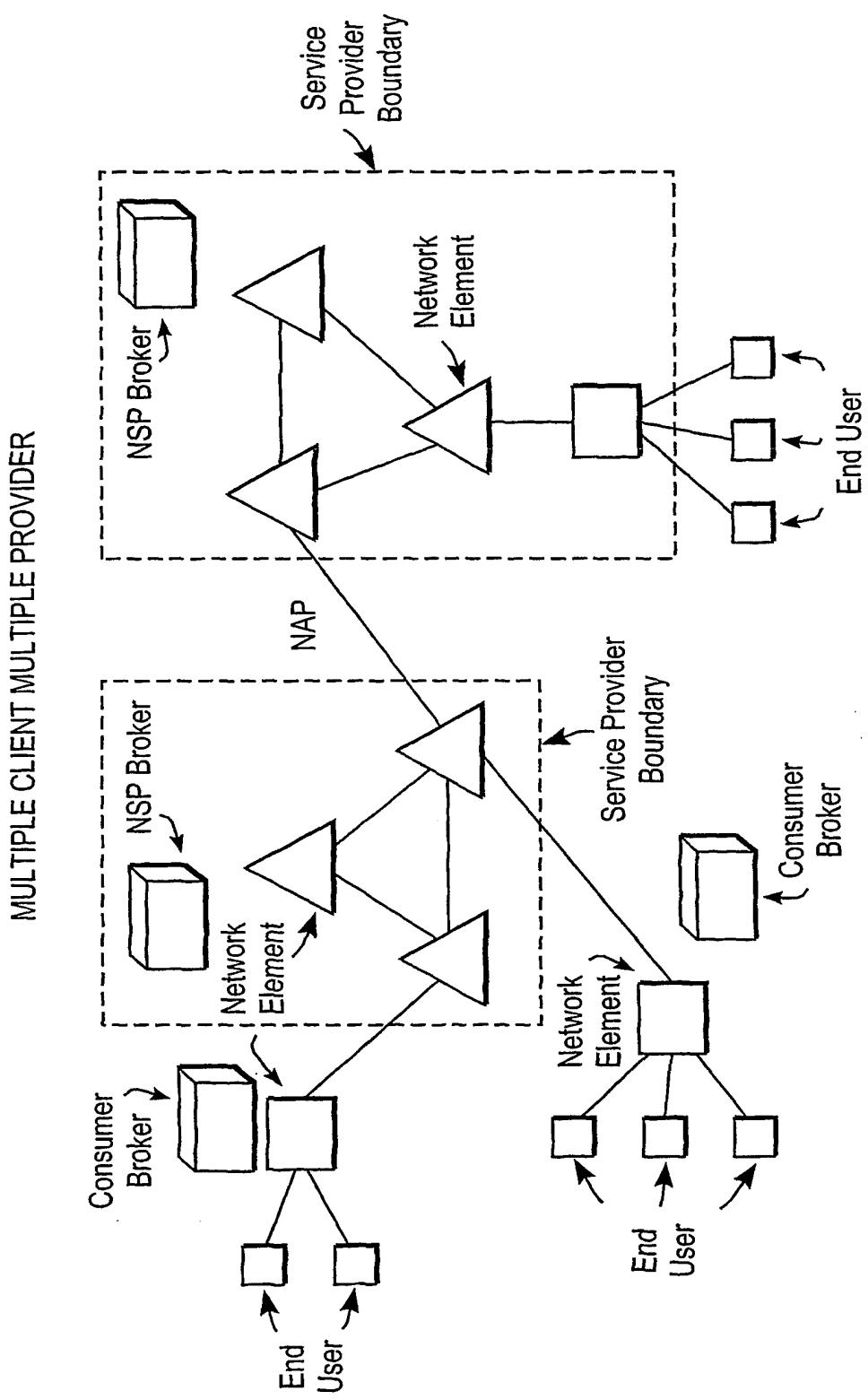


FIG. 31F

35/35

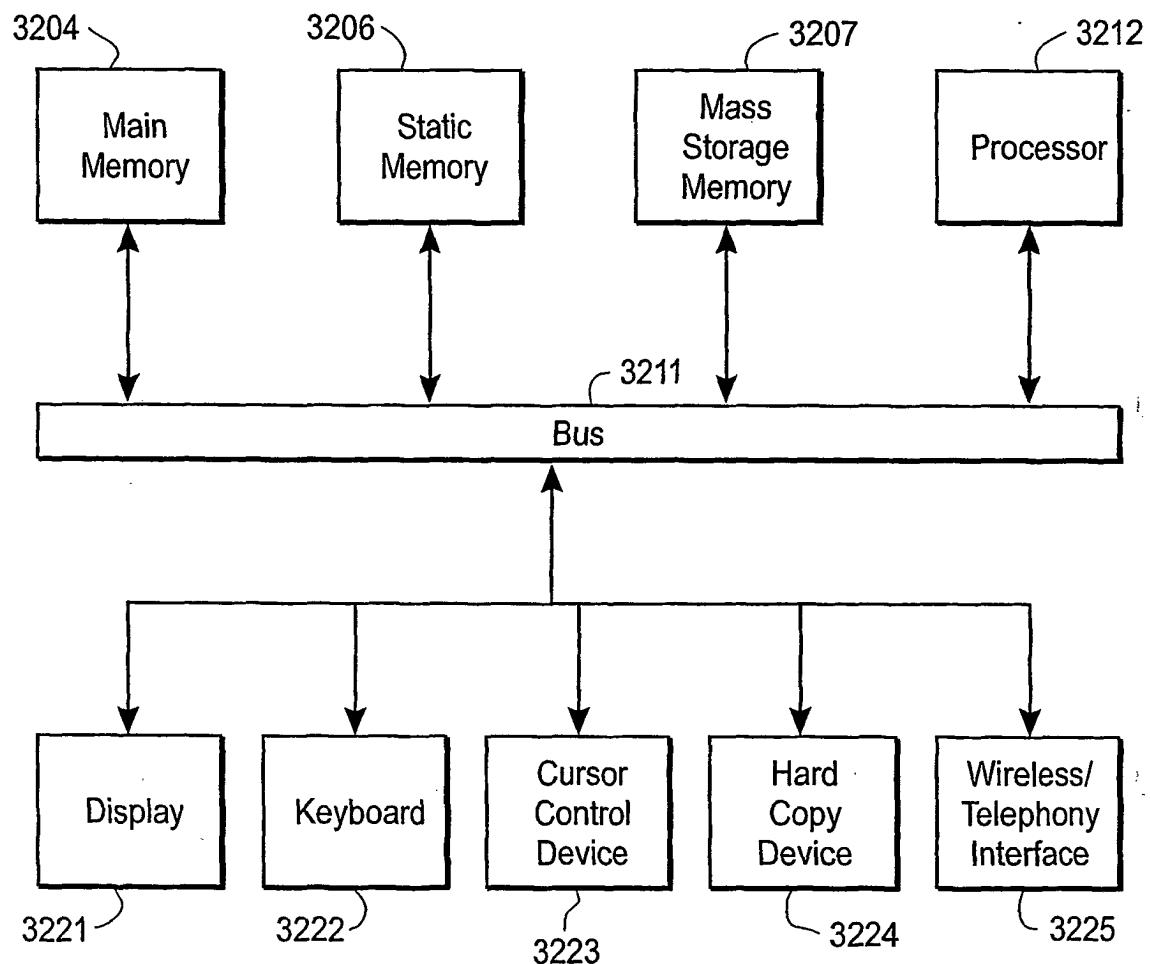
3200

FIG. 32

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/35155

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 13/00

US CL :709/225, 226, 227

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/203, 217, 219, 223, 224, 225, 226, 227, 229, 328, 329

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,295,139 A (PALMER) 15 March, 1994, cols 2-7.	1-79
Y	US 5,604,731 A (GROSSGLAUSER et al) 18 February 1997, cols 3-7.	1-79
A	US 5,745,480 A (BEHTASH et al) 28 April 1998.	1-79
A	US 5,933,816 A (ZEANAH et al) 03 August 1999.	1-79

 Further documents are listed in the continuation of Box C. See patent family annex.

"A"	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	"&"	document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

07 MARCH 2001

Date of mailing of the international search report

9 APR 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

VIET VU

Telephone No. (703) 305-9600